HU-EP-08/30
SFB/CPP-08-61

Automatic generation of Feynman rules in the Schrödinger functional

Shinji Takeda
Humboldt Universität zu Berlin,
Newtonstr. 15, 12489 Berlin, Germany.

August 22, 2008

Abstract

We provide an algorithm to generate vertices for the Schrödinger functional with an abelian background gauge field. The background field has a non-trivial color structure, therefore we mainly focus on a manipulation of the color matrix part. We propose how to implement the algorithm especially in python code. By using python outputs produced by the code, we also show how to write a numerical expression of vertices in the time-momentum as well as the coordinate space into a Feynman diagram calculation code. As examples of the applications of the algorithm, we provide some one-loop results, ratios of the Λ parameters between the plaquette gauge action and the improved gauge actions composed from six-link loops (rectangular, chair and parallelogram), the determination of the O(a) boundary counter term to this order, and the perturbative cutoff effects of the step scaling function of the Schrödinger functional coupling constant.

1 Introduction

It is well known that vertices in lattice gauge theory are quite complicated especially for the pure gauge sector. This is because, on the lattice, while gauge invariance is preserved exactly at a finite lattice spacing, the lattice regularization itself breaks the Lorentz symmetry explicitly. This is the main difficulty of the lattice perturbation theory. To reduce the risk of errors and to alleviate the tedious task of deriving the vertices, it is desirable to have an automatic method. A first attempt was made by Lüscher and Weisz about twenty yeas ago [1]. They worked in momentum space and performed some calculations by using their algorithm which is restricted to closed gauge loops but sufficient for pure gauge theory. About two decades later a new algorithm, which we call bottom up algorithm, was proposed by Hart et. al. [2]. A crucial point in this generalization is that it can deal with any parallel transporter, not only with closed loops. This algorithm also allows us to extend to fermion actions or even smeared HQET actions. In fact, it is applied to the NRQCD [3]. A relevant assumption of this algorithm has been translation

invariance. Needless to say their algorithm also assumes that the background field is set to a trivial one.

Our main concern in this paper is to extend the bottom up algorithm to the Schrödinger functional (SF) [4], where translation invariance for the time direction is broken, and there is an abelian color background gauge field. An essential question here is how to deal with the non-trivial color matrix which involves the complicated background field. Actually this issue can be solved by noting two important properties of the background field, that is, abelian and a linear time dependence. Although the latter property can be relaxed, exploiting this is greatly helpful to gain speed up of the calculations. Our main purpose in this paper is an exposition of the solution. The procedure to generate vertices consists of two parts, first, generating lists which contain information of the non-zero elements of the vertices, and second, writing down a numerical expression of the vertex by making use of the lists as an input. In this paper, we will describe how to write down the vertex in the Feynman diagram calculation code not only for the time-momentum space but also the position space. In the two-loop calculation of the SF coupling [5], it was emphasized that the position space calculation has the advantage of substantially reducing the computational efforts over the time-momentum space. This is the reason why we consider the position space vertex here. As it is the case for the original bottom up algorithm, our generalization to the SF is in principle applicable to both fermion and gauge actions.

The rest of this paper is organized as follows. First we review the bottom up algorithm in the next section. In section 3, after a brief reminder of the SF, we provide a nice representation of link variables and key equations derived from a special property of the background field, and then we show the algorithm to generate lists for the SF. This section is the main part of the paper. Then, an explanation of how to write down the vertices to a program which calculates Feynman diagrams (in particular the Big Mac diagram [5]) is given in section 4. As mentioned before we provide it in both the time-momentum and the position space. As an application, in section 5, we give a one-loop calculation of the SF coupling for various gauge actions including six-link loops. Our conclusion is given in the last section, together with comments on possible future work. Some technical details are given in appendices, like how to partially symmetrize the vertex for the gauge action in section E, further reduction of the number of lists in F, and how to obtain the η derivative of the vertex in G respectively.

2 Bottom up Algorithm

In this section, we summarize the position space version of the bottom up algorithm on the usual translation invariant lattice (non SF). The explanation here is mainly based on [3] apart from the position space. We set up the theory on a hyper-cubic euclidean lattice Λ with spacing a and size $L^3 \times T$. We assume periodic boundary conditions for all directions. A coordinate $x \in \Lambda$ has components x_μ with $\mu = 1, 2, 3, 4$. In the following we denote the link variable (still without background field) by

$$U(x, \mu) = \exp(a g_0 q_\mu(x)), \tag{1}$$

and the anti-hermitian gauge fluctuation field by $q_\mu(x) = \sum_a q^a_\mu(x) T^a$, where T^a are anti-hermitian and are elements of the Lie algebra of $SU(3)$. They are explicitly given in Appendix A.

A first important point for the automatic operation is how to represent the vertices in a program. As an example, we consider a parallel transporter $P(\mathcal{L}, x_{\rm s}, x_{\rm e})$ along a curve $\mathcal{L}$ on the lattice. We denote with $x_{\rm s}$ and $x_{\rm e}$ the start and end point of the parallel transporter. Taylor expansion of the parallel transporter is written as

$$P(\mathcal{L}, x_{\rm s}, x_{\rm e}) = 1 + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{\alpha_1,\cdots,\alpha_r} \sum_{a_1,\cdots,a_r} V^{a_1\cdots a_r}_{\alpha_1\cdots\alpha_r} \prod_{j=1}^{r} q^{a_j}_{\alpha_j}, \tag{2}$$

$$V^{a_1\cdots a_r}_{\alpha_1\cdots\alpha_r} = \mathcal{C}^{a_1\cdots a_r} Y_{\alpha_1\cdots\alpha_r}, \tag{3}$$

where $\alpha = (\mu, x)$ is a combined index labelling links. The (unsymmetrized) vertex $V^{a_1\cdots a_r}_{\alpha_1\cdots\alpha_r}$, has been factorized[1] into the color factor

$$\mathcal{C}^{a_1\cdots a_r} = T^{a_1}\cdots T^{a_r}, \tag{4}$$

and the reduced vertex, $Y_{\alpha_1\cdots\alpha_r}$. The color factor is independent of the shape of the parallel transporter, therefore we consider only the reduced vertex.

For a usual parallel transporter, almost all elements of the corresponding reduced vertex are zero. Therefore usually only non-zero elements are stored in a memory of a machine. This is easily realized as follows. First one labels the non-zero elements of the reduced vertex with number i properly ($i = 1, ..., N_r$), and then one stores its indices, $\alpha_1[i], \cdots, \alpha_r[i]$, and a corresponding non-zero value of the Y, which is represented as the amplitude $f[i]$ in a list

$$L^{(r)}[i] = \{(\alpha_1[i], \cdots, \alpha_r[i]), x_{\rm s}[i], x_{\rm e}[i]; f[i]\}. \tag{5}$$

The number of lists N_r depends on the expansion order r, and of course on the shape of the parallel transporter which one considers. At this point the reader might think $x_{\rm s}$ and $x_{\rm e}$ are redundant but as we will see these are important when dealing with the multiplication of lists. A set of lists of order r, $S^{(r)} = \{L^{(r)}[i] | i = 1, ..., N_r\}$, is "equivalent to" the r'th order reduced vertex,

$$\sum_{\alpha_1,...,\alpha_r} Y_{\alpha_1,...,\alpha_r} q^{a_1}_{\alpha_1} \cdots q^{a_r}_{\alpha_r} = \sum_{i=1}^{N_r} q^{a_1}_{\alpha_1[i]} \cdots q^{a_r}_{\alpha_r[i]} f[i], \tag{6}$$

for a fixed color configuration $\{a_1, \cdots, a_r\}$. Then the parallel transporter in eq.(2) is re-written as

$$P(\mathcal{L}, x_{\rm s}, x_{\rm e}) = 1 + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} \mathcal{C}^{a_1\cdots a_r} \sum_{i=1}^{N_r} q^{a_1}_{\alpha_1[i]} \cdots q^{a_r}_{\alpha_r[i]} f[i]. \tag{7}$$

This expression is useful when deriving the multiplication algorithm as we shall see soon. A set[2] $S = \{S^{(r)} | r = 0, 1, ..., r_{\max}\}$, which represents the reduced vertex up to the order $r_{\max}$ for the parallel transporter, is now encoded in a program.

[1] For the SF, however, this nice structure is rendered more complicated by the presence of the background field as we will see in the next section.

[2] $S^{(r)}$ is a subset of S.

As an explicit example for the list, let us consider the case of a single link variable. From the expanded form

$$U(x,\mu) = 1 + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} \mathcal{C}^{a_1\cdots a_r} q_\alpha^{a_1} \cdots q_\alpha^{a_r}, \tag{8}$$

a corresponding list reads

$$L^{(r)} = \{(\underbrace{\alpha,\cdots,\alpha}_{r \text{ terms}}), x, x + a\hat{\mu}; 1\}, \qquad \alpha = (\mu, x). \tag{9}$$

It consists of only one line ($N_r = 1$) and is the elementary building block of the algorithm, that is, something like an initial condition.

So far we have defined the fundamental elements (the list and set) on which the algorithm operates. Next we discuss how to generate the set S for any parallel transporter. First we assume that two sets $S(\mathcal{L}^{(n)}, x_{\rm s}^{(n)}, x_{\rm e}^{(n)})$ $(n = 1, 2)$ for parallel transporters $P(\mathcal{L}^{(n)}, x_{\rm s}^{(n)}, x_{\rm e}^{(n)})$ are known, and their elements are given by

$$L^{(r)}[i](\mathcal{L}^{(n)}, x_{\rm s}^{(n)}, x_{\rm e}^{(n)}) = \{(\alpha_1^{(n)}[i], \cdots, \alpha_r^{(n)}[i]), x_{\rm s}^{(n)}, x_{\rm e}^{(n)}; f^{(n)}[i]\}. \tag{10}$$

Then, we consider a product of the parallel transporters

$$P(\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda) = P(\mathcal{L}^{(1)}, x_{\rm s}^{(1)}, x_{\rm e}^{(1)}) P(\mathcal{L}^{(2)}, x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda), \tag{11}$$

where $\Lambda = x_{\rm e}^{(1)} - x_{\rm s}^{(2)}$ is a shift vector and is chosen such that the end point of the $n = 1$ parallel transporter and the starting point of the $n = 2$ one are identical. The shift for the $n = 2$ parallel transporter in eq.(11) is taken into account for the list structure as

$$\begin{aligned} L^{(r)}[i](\mathcal{L}^{(2)}, x_{\rm s}^{(2)}, x_{\rm e}^{(2)}) &\stackrel{\Lambda \text{ shift}}{\longrightarrow} L^{(r)}[i](\mathcal{L}^{(2)}, x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda) \\ &= \{(\tilde{\alpha}_1^{(2)}[i], \cdots, \tilde{\alpha}_r^{(2)}[i]), x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda; f^{(2)}[i]\}, \end{aligned} \tag{12}$$

where all x are shifted by Λ and $\tilde{\alpha} = (\mu, x + \Lambda)$. The question now is how to obtain the set for the product of the parallel transporter in eq.(11),

$$S(\mathcal{L}^{(1)}, x_{\rm s}^{(1)}, x_{\rm e}^{(1)}) * S(\mathcal{L}^{(2)}, x_{\rm s}^{(2)}, x_{\rm e}^{(2)}) \longrightarrow S(\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda), \tag{13}$$

from the two given sets $S(\mathcal{L}^{(n)}, x_{\rm s}^{(n)}, x_{\rm e}^{(n)})$ with $n = 1, 2$. Since any parallel transporter is composed of the elementary single link variables, by starting from a set for the link variable whose elements are shown in eq.(9) and by repeating the above multiplication one can obtain a set for arbitrary parallel transporters. This is the origin of the name "bottom up". The algorithm can be understood by looking at the actual multiplication of two parallel transporters. The right

hand side of eq.(11) is expanded as

$$\begin{aligned}
&P(\mathcal{L}^{(1)}, x_{\rm s}^{(1)}, x_{\rm e}^{(1)}) P(\mathcal{L}^{(2)}, x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda) \\
&= 1 + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} \mathcal{C}^{a_1\cdots a_s} \mathcal{C}^{a_{s+1}\cdots a_r} \\
&\times \sum_{s=0}^{r} \frac{r!}{s!(r-s)!} \sum_{i=1}^{N_s^{(1)}} \sum_{j=1}^{N_{r-s}^{(2)}} q^{a_1}_{\alpha_1^{(1)}[i]} \cdots q^{a_s}_{\alpha_s^{(1)}[i]} q^{a_{s+1}}_{\tilde{\alpha}_1^{(2)}[j]} \cdots q^{a_r}_{\tilde{\alpha}_{r-s}^{(2)}[j]} f^{(1)}[i] f^{(2)}[j] \\
&= 1 + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} \mathcal{C}^{a_1\cdots a_r} \sum_{k=1}^{N_r} q^{a_1}_{\alpha_1[k]} \cdots q^{a_s}_{\alpha_s[k]} q^{a_{s+1}}_{\tilde{\alpha}_{s+1}[k]} \cdots q^{a_r}_{\alpha_r[k]} f[k]. \qquad (14)
\end{aligned}$$

After the first equal-sign we inserted eq.(7). In the last step, we have used the fact that the color factor is independent of the shape of the parallel transporter and have combined the three summations (over s, i and j) into that over k. In short, we did a relabelling of the indices, and rewrote the amplitude factor. Finally the resulting lists $L^{(r)}[k](\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda)$ are created by putting the new label structure and the new amplitude. The algorithm is summarized as follows

- Relabelling:

$$\{\alpha_1^{(1)}[i], \cdots, \alpha_s^{(1)}[i], \tilde{\alpha}_1^{(2)}[j], \cdots, \tilde{\alpha}_{r-s}^{(2)}[j]\} \longrightarrow \{\alpha_1[k], \cdots, \alpha_s[k], \alpha_{s+1}[k], \cdots, \alpha_r[k]\}, \qquad (15)$$

- Amplitude part:

$$\frac{r!}{s!(r-s)!} f^{(1)}[i] f^{(2)}[j] \longrightarrow f[k], \qquad (16)$$

- Creating list:

$$\begin{aligned}
&L^{(s)}[i](\mathcal{L}^{(1)}, x_{\rm s}^{(1)}, x_{\rm e}^{(1)}) * L^{(r-s)}[j](\mathcal{L}^{(2)}, x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda) \longrightarrow \\
&\quad L^{(r)}[k](\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda) = \{(\alpha_1[k], \cdots, \alpha_r[k]), x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda; f[k]\}. \qquad (17)
\end{aligned}$$

This procedure should be carried out for $0 \le s \le r$, $1 \le i \le N_s^{(1)}$ and $1 \le j \le N_{r-s}^{(2)}$ if order r is desired. This is an algorithm to obtain the set $S(\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda)$ and has been implemented in the python script language, which has a great capability of dealing with complicated list operations. In this way, one can obtain the vertices for any parallel transporter.

3 Extension to the Schrödinger functional

Let us now proceed to the SF [6]. This section is the central part of the paper.

3.1 Preliminary

We consider, in the following section, a finite box of size $L^3 \times T$ with periodic boundary conditions in the spatial directions. We impose Dirichlet boundary conditions for the link variables at the time boundaries,

$$U(x,k)|_{x_4=0} = \exp\{aC\}, \qquad U(x,k)|_{x_4=T} = \exp\{aC'\}, \tag{18}$$

where the boundary fields C, C' in the SU(3) gauge theory [4] are given in Appendix B. The SF in the theory is defined as

$$\mathcal{Z}[C,C'] = \int D[U] e^{-S[U]}. \tag{19}$$

It is shown in Ref. [6] that a minimum of the plaquette gauge action is given by the lattice background field $U(x,\mu) = V(x,\mu)$, whose elements are given by

$$V(x,4) = 1, \quad V(x,k) = V(x_4) = e^{ia(\mathcal{E}x_4 - iC)}, \tag{20}$$

where $V(x,k)$ is independent of the spatial direction k. An explicit form of the color electric field $\mathcal{E}$ is given in Appendix B. For later use, we introduce a convenient notation for the background field

$$V(x,\mu) = (V(x_4))^{\sigma_\mu} = \begin{cases} V(x_4)^0 = 1, & \text{for } \mu = 4, \\ V(x_4)^1 = V(x_4), & \text{for } \mu = 1,2,3, \end{cases} \tag{21}$$

$$\sigma_\mu = 1 - \delta_{\mu 4} = (1,1,1,0). \tag{22}$$

By using these notations, we can express the classical link variable as

$$V(x,\mu)^m = V(x_4)^{m\sigma_\mu}, \tag{23}$$

where m takes the values ± 1 and $V(x,\mu)^m$ represents $V(x,\mu)$ for $m = 1$, and $V(x,\mu)^\dagger = V(x,\mu)^{-1}$ for $m = -1$.

3.2 An expression for the link variable

In this section, we introduce a proper expression for the link variable $U(x,\mu)$ in the SF case which will be useful when generating vertices. In the presence of a background field, the link variable is written as

$$U(x,\mu) = e^{ag_0 q_\mu(x)} V(x,\mu). \tag{24}$$

When dealing with the SF in the presence of the abelian background field, it is convenient to use a color decomposition,

$$q_4(x) = \sum_{a=1}^{8} q_4^a(x) I^a, \tag{25}$$

$$q_k(x) = \sum_{a=1}^{8} q_k^a(x) I^a e^{i\phi_a(x_4)/2}, \tag{26}$$

a	$\phi_a(x_4)$	ϕ'_a	$\partial_\eta \phi_a(x_4)$	$\partial_\eta \phi'_a$
1	$-3a\gamma x_4 + \frac{a}{L}(\eta[\frac{3}{2} - \nu] - \frac{\pi}{3})$	$-3a^2\gamma$	$-3ax_4/LT + \frac{a}{L}[\frac{3}{2} - \nu]$	$-3a^2/LT$
4	$-3a\gamma x_4 + \frac{a}{L}(\eta[\frac{3}{2} + \nu] - \frac{2\pi}{3})$	$-3a^2\gamma$	$-3ax_4/LT + \frac{a}{L}[\frac{3}{2} + \nu]$	$-3a^2/LT$
3	0	0	0	0
6	$\frac{a}{L}(2\eta\nu - \frac{\pi}{3})$	0	$\frac{a}{L}2\nu$	0
8	0	0	0	0

Table 1: Phases $\phi_a(x_4)$, ϕ'_a, $\partial_\eta \phi_a(x_4)$ and $\partial_\eta \phi'_a$ for SU(3) case. The other components $a = 2, 5, 7$, which are not shown here, are defined as $\phi_2 = -\phi_1$, $\phi_5 = -\phi_4$, $\phi_7 = -\phi_6$. The γ is given in eq.(133) in Appendix B.

where an overview of the phases $\phi_a(x_4)$ are given in Table 1. The idea of inclusion of the phase can be found in the original SF paper [6]. The basis of the Lie algebra I^a can be found in [7, 8] and is also summarized in Appendix A. By using σ_μ in eq.(22), the expression for $q_4(x)$ and $q_k(x)$ are unified as

$$q_\mu(x) \;=\; \sum_{a=1}^{8} q_\mu^a(x) I^a e^{i\sigma_\mu \phi_a(x_4)/2}. \tag{27}$$

By using the above expression, the link variable in eq.(24) can be written as

$$U(x,\mu) = V(x_4)^{\sigma_\mu} + \sum_{r=1}^{\infty} \frac{(ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} q_\alpha^{a_1} \cdots q_\alpha^{a_r} \left[I^{a_1} \cdots I^{a_r} V(x_4)^{\sigma_\mu}\right] e^{i\sigma_\mu \sum_{u=1}^r \phi_{a_u}(x_4)/2}, \tag{28}$$

where we have used eq.(23) for the lattice background field and introduced the notation $[II\cdots I]$ for representing a 3×3 color matrix.

Now we will derive an expression for $U(x,\mu)^\dagger = U(x,\mu)^{-1}$. Thanks to the phase factor introduced in eq.(28), it will turn out that it has a form similar to U. Let us show this explicitly. The inverse link variable can be expanded in the following way

$$\begin{aligned} U(x,\mu)^{-1} \;&=\; V(x,\mu)^{-1} e^{-ag_0 q_\mu(x)} = V(x_4)^{-\sigma_\mu} \\ &+\; \sum_{r=1}^{\infty} \frac{(-ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} q_\alpha^{a_1} \cdots q_\alpha^{a_r} \left[V(x_4)^{-\sigma_\mu} I^{a_1} \cdots I^{a_r}\right] e^{i\sigma_\mu \sum_{u=1}^r \phi_{a_u}(x_4)/2}. \end{aligned} \tag{29}$$

This naive form is not suited for our purpose. The location of the V is different from the one in eq.(28). We want to put V to very right in the color matrix. For this purpose, we employ key formulas which will play an important role in the following

$$\left[V(x_4) I^a V(x_4)^{-1}\right] \;=\; [I^a]\, e^{i\phi_a(x_4)}, \tag{30}$$

$$\left[e^{ia^2\mathcal{E}} I^a e^{-ia^2\mathcal{E}}\right] \;=\; [I^a]\, e^{i\phi'_a}, \tag{31}$$

where ϕ'_a are shown in Table 1. The formulas are derived from a special property of the Abelian background field whose generators are elements of the Cartan sub-algebra. More details on the

formulas are given in Appendix C. Let us return to U^{-1}. By applying the formula in eq.(30) to eq.(29), we obtain the following expression

$$\begin{aligned} U(x,\mu)^{-1} &= V(x_4)^{-\sigma_\mu} \\ &+ \sum_{r=1}^{\infty} \frac{(-ag_0)^r}{r!} \sum_{a_1,\cdots,a_r} q_\alpha^{a_1}\cdots q_\alpha^{a_r} \left[I^{a_1}\cdots I^{a_r} V(x_4)^{-\sigma_\mu}\right] e^{-i\sigma_\mu \sum_{u=1}^r \phi_{a_u}(x_4)/2}, \end{aligned} \tag{32}$$

where we have been able to place V in very right in the color matrix. Note the sign of the phase. Now, we can express the U and U^{-1} in a unified form, for $m = \pm 1$,

$$\begin{aligned} U(x,\mu)^{m} &= V(x_4)^{m\sigma_\mu} \\ &+ \sum_{r=1}^{\infty} \frac{(mag_0)^r}{r!} \sum_{a_1,\cdots,a_r} q_\alpha^{a_1}\cdots q_\alpha^{a_r} \left[I^{a_1}\cdots I^{a_r} V(x_4)^{m\sigma_\mu}\right] e^{im\sigma_\mu \sum_{u=1}^r \phi_{a_u}(x_4)/2}. \end{aligned} \tag{33}$$

Note the location of m.

In the following the term 'color factor' is reserved for the color matrix part multiplied with the phase factors. The corresponding color factor for the single link variable reads

$$\left[I^{a_1} I^{a_2}\cdots I^{a_r} V(x_4)^{m\sigma_\mu}\right] e^{im\sigma_\mu \sum_{u=1}^r \phi_{a_u}(x_4)/2}. \tag{34}$$

An essential new ingredient for the case of the SF is this color factor which depends on x_4, m and μ, while a color factor with the zero background field case $[I^{a_1} I^{a_2}\cdots I^{a_r}]$ is independent of them.

3.3 Multiplication of the color factors

Before entering the case of the SF, it is maybe worth discussing the color factor in the vanishing background field case in order to make the difference between the two cases clear. As mentioned in Section 2, the color factor in the vanishing background field case is independent of the shape of the path, therefore the multiplication of color factors is trivial

$$\mathcal{C}^{a_1\cdots a_s}\mathcal{C}^{a_{s+1}\cdots a_r} = \mathcal{C}^{a_1\cdots a_s a_{s+1}\cdots a_r}. \tag{35}$$

On the other hand, for the SF case, due to the presence of the non-vanishing background field, the color factor depends on the shape of the path and its multiplication is not as trivial as shown above. In this section we will present a multiplication rule for the color factor for the SF.

Our main finding is that any color factor in the SF of order r can be cast in the form

$$\mathcal{C}^{a_1\cdots a_r}(x_{l,4}, A, B, C, D) = \underbrace{\left[I^{a_1}\cdots I^{a_r} V(x_{l,4})^A e^{ia^2\mathcal{E}B}\right]}_{3\times 3 \text{ matrix}} \underbrace{e^{\frac{i}{2}\sum_{u=1}^r (C_u\phi'_{a_u} + D_u\phi_{a_u}(x_{l,4}))}}_{U(1) \text{ phase}}. \tag{36}$$

This can be shown by making use of two properties, first, the background field being abelian eq.(30,31), and second, the linear time dependence[3] of the exponent of $V(x_4)$ in eq.(20). Note

[3] In some special case like [9], the property of the linear time dependence is lost. A multiplication algorithm for this case is given in Appendix D, and it has already been used in a one-loop calculation in the reference.

that the color factor has a time dependence which comes from the background field. We choose it as $x_{l,4}$ which is the time component of the coordinate of the left-most link variable of the parallel transporter

$$P(\mathcal{L}, x_{\rm s}, x_{\rm e}) = U(x_l, \mu)^m U U \cdots , \tag{37}$$

with $x_{\rm s} = x_l + a\hat{\mu}(1-m)/2$. In practice, x_l is set to the origin. The components A and B are single component integer. C and D are multi-component integers of size r

$$\begin{aligned} C &= (C_1, \cdots, C_r), & (38)\\ D &= (D_1, \cdots, D_r). & (39) \end{aligned}$$

The ϕ'_a in eq.(36) are the time derivative of the $\phi_a(x_4)$ and do them-self not depend on the time. Their values are given in Table 1. In the expression of the color factor, the information about the lattice size and the background field parameters (η, ν) is encoded in $V(x_{l,4})$, $\mathcal{E}$, and the phases $\phi_a(x_{l,4})$, ϕ'_a. When producing lists, this information is actually irrelevant, only $x_{l,4}$, A, B, C and D are required. The former is only needed when writing down a numerical expression of a vertex to a diagram calculation program at a second stage. The benefit of the expression is that we can separate the information $x_{l,4}$, A, B, C and D (which will be included into the list structure) and the lattice structure information. Therefore, we can perform a symbolic list operation, independently of the detail of the lattice size and the background field parameters.

Let us give an example of a configuration of $(x_{l,4}, A, B, C, D)$ for the single link $U(x,\mu)^m$ whose color factor is given in eq.(34)

$$\begin{aligned} x_{l,4} &= x_4, & (40)\\ A &= m\sigma_\mu, & (41)\\ B &= 0, & (42)\\ C &= (\underbrace{0, \cdots, 0}_{r \text{ terms}}), & (43)\\ D &= (\underbrace{m\sigma_\mu, \cdots, m\sigma_\mu}_{r \text{ terms}}). & (44) \end{aligned}$$

We have obtained a manageable expression for the color factor. Next we formulate a multiplication for the integer list $(x_{l,4}, A, B, C, D)$. An important point is that even in the SF the multiplication of the color factor is closed[4].

$$\begin{aligned} \mathcal{C}^{a_1\cdots a_s}(x^{(1)}_{l,4}, A^{(1)}, B^{(1)}, C^{(1)}, D^{(1)}) \;\times\; & \mathcal{C}^{a_{s+1}\cdots a_r}(x^{(2)}_{l,4}, A^{(2)}, B^{(2)}, C^{(2)}, D^{(2)}) \\ = \; & \mathcal{C}^{a_1\cdots a_r}(x_{l,4}, A, B, C, D). \end{aligned} \tag{45}$$

From an actual multiplication of the color factors, we find the algorithm to get a corresponding list from two given lists $(x^{(1)}_{l,4}, A^{(1)}, B^{(1)}, C^{(1)}, D^{(1)})$ and $(x^{(2)}_{l,4}, A^{(2)}, B^{(2)}, C^{(2)}, D^{(2)})$ in the

[4]This can be understood from the background field gauge transformation in Ref. [10]

following way

$$
\begin{aligned}
x_{l,4} &\longleftarrow x^{(1)}_{l,4}, && (46)\\
A &\longleftarrow A^{(1)} + A^{(2)}, && (47)\\
B &\longleftarrow B^{(1)} + B^{(2)} + \Delta t A^{(2)}, && (48)\\
C &\longleftarrow \underbrace{(\underbrace{C^{(1)}_1, \cdots, C^{(1)}_s}_{s \text{ terms}}, \underbrace{C^{(2)}_1 + 2B^{(1)} + \Delta t D^{(2)}_1, \cdots, C^{(2)}_{r-s} + 2B^{(1)} + \Delta t D^{(2)}_{r-s}}_{(r-s) \text{ terms}})}_{r \text{ terms}}, && (49)\\
D &\longleftarrow \underbrace{(\underbrace{D^{(1)}_1, \cdots, D^{(1)}_s}_{s \text{ terms}}, \underbrace{D^{(2)}_1 + 2A^{(1)}, \cdots, D^{(2)}_{r-s} + 2A^{(1)}}_{(r-s) \text{ terms}})}_{r \text{ terms}}, && (50)
\end{aligned}
$$

where $\Delta t = x^{(2)}_{l,4} - x^{(1)}_{l,4}$. It turns out that the resulting A and B remain single component integers, on the other hand, the resulting C (D) is given by combining $C^{(1)}$ ($D^{(1)}$) and $C^{(2)}$ ($D^{(2)}$) with some additional terms for the latter part. Since $x_{l,4}$, A, B, C and D are all integer values and this operation is simple, the multiplication algorithm is suited for a symbolic operation and is easily implemented in the python script language.

In an actual implementation, as a new ingredient, we have to add the new components x_l, A, B, C and D to the earlier list structure,

$$
\begin{aligned}
L^{(r)}[k] &= \{(\alpha_1[k], \cdots, \alpha_r[k]), x_{\rm s}, x_{\rm e}, ; f[k]\}\\
&\rightarrow \{(\alpha_1[k], \cdots, \alpha_r[k]), x_{\rm s}, x_{\rm e}, x_l, A[k], B[k],\\
&\quad (C_1[k], \cdots, C_r[k]), (D_1[k], \cdots, D_r[k]); f[k]\}. \qquad (51)
\end{aligned}
$$

For two given lists, $n = 1, 2$,

$$
\begin{aligned}
L^{(r)}[i](\mathcal{L}^{(n)}, x^{(n)}_{\rm s}, x^{(n)}_{\rm e}) &= \{(\alpha^{(n)}_1[i], \cdots, \alpha^{(n)}_r[i]), x^{(n)}_{\rm s}, x^{(n)}_{\rm e}, x^{(n)}_l, A^{(n)}[i], B^{(n)}[i],\\
&\quad (C^{(n)}_1[i], \cdots, C^{(n)}_r[i]), (D^{(n)}_1[i], \cdots, D^{(n)}_r[i]); f^{(n)}[i]\}, \qquad (52)
\end{aligned}
$$

the multiplication algorithm for lists is summarized as follows (the relabelling and amplitude parts are the same as before).

- Relabelling:

$$
\{\alpha^{(1)}_1[i], \cdots, \alpha^{(1)}_s[i], \tilde{\alpha}^{(2)}_1[j], \cdots, \tilde{\alpha}^{(2)}_{r-s}[j]\} \longrightarrow \{\alpha_1[k], \cdots, \alpha_s[k], \alpha_{s+1}[k], \cdots, \alpha_r[k]\}, \quad (53)
$$

- Color factor part (eq.(46)-(50)):

$$
\begin{aligned}
(x^{(1)}_{l,4}, A^{(1)}[i], B^{(1)}[i], C^{(1)}[i], D^{(1)}[i]) \quad &* \quad (x^{(2)}_{l,4}, A^{(2)}[j], B^{(2)}[j], C^{(2)}[j], D^{(2)}[j])\\
&\longrightarrow (x_{l,4}, A[k], B[k], C[k], D[k]), \qquad (54)
\end{aligned}
$$

- Amplitude part:

$$\frac{r!}{s!(r-s)!} f^{(1)}[i] f^{(2)}[j] \longrightarrow f[k], \tag{55}$$

- Creating the list:

$$\begin{aligned} & L^{(s)}[i](\mathcal{L}^{(1)}, x_{\rm s}^{(1)}, x_{\rm e}^{(1)}) * L^{(r-s)}[j](\mathcal{L}^{(2)}, x_{\rm s}^{(2)} + \Lambda, x_{\rm e}^{(2)} + \Lambda) \\ \longrightarrow \quad & L^{(r)}[k](\mathcal{L}^{(1)} * \mathcal{L}^{(2)}, x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda) \\ = \quad & \{(\alpha_1[k], \cdots, \alpha_r[k]), x_{\rm s}^{(1)}, x_{\rm e}^{(2)} + \Lambda, x_l, A[k], B[k], \\ & (C_1[k], \cdots, C_r[k]), (D_1[k], \cdots, D_r[k]); f[k]\}. \end{aligned} \tag{56}$$

In order to start the multiplication algorithm, we need an initial set of lists. It is the set for a single link variable $U(x,\mu)^m$ which is given by the list

$$L^{(r)} = \{(\underbrace{\alpha, \cdots, \alpha}_{r \text{ terms}}), x_{\rm s}, x_{\rm e}, x_l, m\sigma_\mu, 0, (\underbrace{0, \cdots, 0}_{r \text{ terms}}), (\underbrace{m\sigma_\mu, \cdots, m\sigma_\mu}_{r \text{ terms}}); m^r\}, \tag{57}$$

with

$$x_{\rm s} = x + a\hat{\mu}\frac{1-m}{2}, \tag{58}$$

$$x_{\rm e} = x + a\hat{\mu}\frac{1+m}{2}, \tag{59}$$

$$x_l = x. \tag{60}$$

To check that the algorithm works properly, we perform a one-loop calculation of the SF coupling. The results are shown in Section 5. By calculating this quantity, we can check the two-point vertex. We investigate not only the plaquette gauge action but also the improved gauge actions including six-link loops. As a further check we compute the Big Mac diagram, which includes three-point vertex, for smaller lattices and check the consistency with the old results (private communication with Peter Weisz).

4 How to exploit the set for the gauge action

In this section, by using the python output (set S) generated by the above algorithm, we describe how to implement the vertex in a code which calculates Feynman diagrams. We exclusively consider the gauge actions, and in both coordinate space and time-momentum space for future purposes. In this section we write μ and x explicitly instead of using the super index $\alpha = (\mu, x)$, and set the lattice unit $a = 1$.

4.1 Coordinate space representation

The gauge action[5] consists of a loop $\mathcal{C}$

$$S_{\rm G} = \frac{2}{g_0^2}\sum_{\mathcal{C}} {\rm Retr}[1 - U(\mathcal{C})], \tag{61}$$

where $U(\mathcal{C})$ is the parallel transporter around $\mathcal{C}$. Its expansion in terms of g_0 in the coordinate space is given by

$$S_{\rm G} = \sum_{r=2}^{\infty} \frac{(g_0)^{r-2}}{r!} \sum_{x_1,\cdots,x_r} \sum_{\mu_1,\cdots,\mu_r} \sum_{a_1,\cdots,a_r} V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r) \prod_{j=1}^{r} q^{a_j}_{\mu_j}(x_j). \tag{62}$$

The symmetrized vertex V is given by summation over unsymmetrized vertices Y

$$V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r) = \frac{1}{r!}\sum_{\sigma\in\mathcal{S}_r} \sigma\cdot Y^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r), \tag{63}$$

where $\mathcal{S}_r$ is the permutation group of the order r. The symmetrized vertex may be rewritten as

$$V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r) = \frac{1}{r!}\sum_{\sigma\in\mathcal{S}_r/\mathcal{Z}_r} \sigma\cdot Y'^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r), \tag{64}$$

where the partially symmetrized vertex is

$$Y'^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r) = \sum_{\sigma\in\mathcal{Z}_r} \sigma\cdot Y^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(x_1,\cdots,x_r). \tag{65}$$

$\mathcal{Z}_r$ is defined as a subset of the total permutation group $\mathcal{S}_r$ and consists of the inversion and cyclic permutations. The summation over $\mathcal{Z}_r$ in eq.(65) can be done at the python level even for the SF. Some details about the partial symmetrization is given in Appendix E. The number of elements of $\mathcal{S}_r$ and $\mathcal{Z}_r$ are $\#\mathcal{S}_r = r!$ and $\#\mathcal{Z}_r = 2r$ respectively. Therefore there are remaining permutations $\#\mathcal{S}_r/\mathcal{Z}_r = r!/2r = (r-1)!/2$. Note that for $r \geq 4$ we need to perform the remaining symmetrization (summation in eq.(64)) at the Feynman diagram calculation stage.

To make the discussion more concrete, we use the plaquette gauge action as a specific example. First we consider a sum of the plaquette over all directions at fixed point

$$\sum_{\mu>\nu} {\rm Retr}\left[U(0,\mu)U(0+\hat{\mu},\nu)U(0+\hat{\nu},\mu)^\dagger U(0,\nu)^\dagger\right], \tag{66}$$

and the corresponding set whose elements are lists of the from

$$\begin{aligned} L^{(r)}[i] &= \{(\mu_1[i],\cdots,\mu_r[i]),(dx_1[i],\cdots,dx_r[i]),0,0,0,A[i],B[i], \\ &\quad (C_1[i],\cdots,C_r[i]),(D_1[i],\cdots,D_r[i]);f[i]\}, \end{aligned} \tag{67}$$

[5] We will give more precise definition of the gauge action in Section 5. Here the reality and trace properties are important.

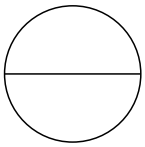

Figure 1: The Big Mac diagram

where we assume that the partial symmetrization procedure has been done. As described in Ref. [1], by making use of the translation invariance in the spatial directions, the $d\mathbf{x}_1, ..., d\mathbf{x}_r$ are translated to the standard form [2], so that the number of lists can be reduced. On the other hand, this reduction does not apply for the time direction due to the loss of the time translation invariance in the SF. Note that in eq.(66) the summation over the coordinates is not taken but kept. In this case we may set $x_{\rm s} = x_{\rm e} = x_l = 0$. If one carries out the sum at the python level, it renders the number of lists prohibitively large of order L^3T, and this is apparently inefficient. Instead the sum is carried out in the diagram calculation code. In order to obtain the complete vertex, we have to loop over the x_l exhausting the whole elements of the vertex. In this way, all non-zero elements of the vertex are assigned.

Let us show how to use the the set for the partially symmetrized vertex in a diagram calculation code. As an example, let us take the Big Mac diagram, which appears in the two-loop SF coupling calculation [5], shown in Figure 1, whose expression in the coordinate space is given by

$$\begin{aligned} \text{BigMac} &= \frac{1}{12}\sum_{\{a\},\{\mu\},\{x\}} V^{a_1a_2a_3}_{\mu_1\mu_2\mu_3}(x_1,x_2,x_3)V^{a_4a_5a_6}_{\mu_4\mu_5\mu_6}(x_4,x_5,x_6) \\ &\quad\times\langle q^{a_1}_{\mu_1}(x_1)q^{a_4}_{\mu_4}(x_4)\rangle_0\langle q^{a_2}_{\mu_2}(x_2)q^{a_5}_{\mu_5}(x_5)\rangle_0\langle q^{a_3}_{\mu_3}(x_3)q^{a_6}_{\mu_6}(x_6)\rangle_0, \end{aligned} \tag{68}$$

where $1/12$ is a symmetry factor and V is the symmetrized three-point gluon vertex. The free gluon propagator in the coordinate space is expressed in terms of that in the time-momentum space $D^a_{\mu\nu}(\mathbf{p}, x_0, y_0)$, which gauge fixing is done,

$$\langle q^a_\mu(x)q^b_\nu(y)\rangle_0 = \delta_{b\bar{a}}S^a_{\mu\nu}(x_0,y_0,\mathbf{x},\mathbf{y}), \tag{69}$$

and

$$\begin{aligned} S^a_{\mu\nu}(x_0,y_0,\mathbf{x},\mathbf{y}) &= S^a_{\mu\nu}(x_0,y_0,\mathbf{x}-\mathbf{y}) \\ &= \frac{1}{L^3}\sum_{\mathbf{p}} e^{i\mathbf{p}(\mathbf{x}-\mathbf{y}+\frac{1}{2}\hat{\mu}\sigma_\mu-\frac{1}{2}\hat{\nu}\sigma_\nu)}D^a_{\mu\nu}(\mathbf{p},x_0,y_0). \end{aligned} \tag{70}$$

The notation $\bar{a}$ is defined in Appendix A. In the following, we use the notation $t = x_4$. By using the propagator S and performing the partial sum over the color index a_4, a_5 and a_6, the BigMac diagram is given by

$$\begin{aligned} \text{BigMac} &= \frac{1}{12}\sum_{a_1,a_2,a_3}\sum_{\{\mu\},\{t\},\{\mathbf{x}\}} V^{a_1a_2a_3}_{\mu_1\mu_2\mu_3}(t_1,t_2,t_3,\mathbf{x}_1,\mathbf{x}_2,\mathbf{x}_3)V^{\bar{a_1}\bar{a_2}\bar{a_3}}_{\mu_4\mu_5\mu_6}(t_4,t_5,t_6,\mathbf{x}_4,\mathbf{x}_5,\mathbf{x}_6) \\ &\quad\times S^{a_1}_{\mu_1\mu_4}(t_1,t_4,\mathbf{x}_1-\mathbf{x}_4)S^{a_2}_{\mu_2\mu_5}(t_2,t_5,\mathbf{x}_2-\mathbf{x}_5)S^{a_3}_{\mu_3\mu_6}(t_3,t_6,\mathbf{x}_3-\mathbf{x}_6). \end{aligned} \tag{71}$$

The color factor of order r for the gauge action is given by

$$\begin{aligned}&\mathcal{C}_{\rm G}^{a_1\cdots a_r}(t,A,B,C,D)\\ &= \ {\rm tr}[I^{a_1}\cdots I^{a_r}e^{i\mathcal{E}B}+(-1)^r I^{a_r}\cdots I^{a_1}e^{-i\mathcal{E}B}]e^{\frac{i}{2}\sum_{u=1}^{r}(C_u\phi'_{a_u}+D_u\phi_{a_u}(t))}.\end{aligned} \tag{72}$$

Actually almost all elements of the color factor are zero, therefore it is convenient to use a color index

$$(a_1[n],a_2[n],a_3[n]), \tag{73}$$

with running index n which contains indices for the nonzero elements of the color factor. Furthermore, another list structure is used for the vertex. The symmetrized vertex of order $r=3$ is written as

$$V^{a_1a_2a_3}_{\mu_1\mu_2\mu_3}(x_1,x_2,x_3)=\frac{1}{3!}Y'^{a_1a_2a_3}_{\mu_1\mu_2\mu_3}(x_1,x_2,x_3), \tag{74}$$

where Y' is the partially symmetrized vertex. At order $r=3$, V and Y' are equivalent up to an overall factor 3!. The partially symmetrized vertex can be generated from the python output, the list structure for the first gluon vertex in eq.(71) is generated from

$$\begin{aligned}L^{(3)}[i]=\{&(\mu_1[i],\mu_2[i],\mu_3[i]),((dt_1[i],d\mathbf{x}_1[i]),(dt_2[i],d\mathbf{x}_2[i]),(dt_3[i],d\mathbf{x}_3[i])),\\ &0,0,0,A[i],B[i],(C_1[i],C_2[i],C_3[i]),(D_1[i],D_2[i],D_3[i]);f[i]\},\end{aligned} \tag{75}$$

with index i identifying the list. The number of lists is $N_3=396$ for the plaquette action. For the second vertex we use the index j for identification. The coordinates t_m and $\mathbf{x}_m$ ($m=1,2,3,4,5,6$) are given in terms of the elements of the list as well as t, s, $\mathbf{x}$ and $\mathbf{y}$,

$$\mathbf{x}_1=\mathbf{x}+d\mathbf{x}_1[i]\,,\qquad t_1=t+dt_1[i], \tag{76}$$
$$\mathbf{x}_2=\mathbf{x}+d\mathbf{x}_2[i]\,,\qquad t_2=t+dt_2[i], \tag{77}$$
$$\mathbf{x}_3=\mathbf{x}+d\mathbf{x}_3[i]\,,\qquad t_3=t+dt_3[i], \tag{78}$$
$$\mathbf{x}_4=\mathbf{y}+d\mathbf{x}_1[j],\qquad t_4=s+dt_1[j], \tag{79}$$
$$\mathbf{x}_5=\mathbf{y}+d\mathbf{x}_2[j],\qquad t_5=s+dt_2[j], \tag{80}$$
$$\mathbf{x}_6=\mathbf{y}+d\mathbf{x}_3[j],\qquad t_6=s+dt_3[j]. \tag{81}$$

Now the indices (t, $\mathbf{x}$) and (s, $\mathbf{y}$) run over the whole lattice. By making use of the above parameterizations and the translation invariance for the spatial direction, the diagram is written

as

$$\begin{aligned}
\text{BigMac} &= \frac{L^3}{12}\sum_{n(\text{color loop})}^{52}\sum_{i,j(\text{python loop})}^{396}\sum_{t,s}\sum_{\mathbf{x}} \\
&\quad \frac{1}{3!}f[i]\mathcal{C}_{\text{G}}^{a_1[n]a_2[n]a_3[n]}(t,A[i],B[i],C[i],D[i]) \\
&\quad \times\frac{1}{3!}f[j]\mathcal{C}_{\text{G}}^{\bar{a}_1[n]\bar{a}_2[n]\bar{a}_3[n]}(s,A[j],B[j],C[j],D[j]) \\
&\quad \times S^{a_1[n]}_{\mu_1[i]\mu_1[j]}(t+dt_1[i],s+dt_1[j],\mathbf{x}+d\mathbf{x}_1[i]-d\mathbf{x}_1[j]) \\
&\quad \times S^{a_2[n]}_{\mu_2[i]\mu_2[j]}(t+dt_2[i],s+dt_2[j],\mathbf{x}+d\mathbf{x}_2[i]-d\mathbf{x}_2[j]) \\
&\quad \times S^{a_3[n]}_{\mu_3[i]\mu_3[j]}(t+dt_3[i],s+dt_3[j],\mathbf{x}+d\mathbf{x}_3[i]-d\mathbf{x}_3[j]). \qquad (82)
\end{aligned}$$

Note that the summations over $\{\mu\}$, $\{t\}$ and $\{\mathbf{x}\}$ in eq.(68) of order $4^6\times T^6(L^3)^6$ are replaced by i,j (python loop), t,s and $\mathbf{x}$ of order $(396)^2\times T^2L^3$. In the summation step, one has to be cautious about the boundary conditions,

$$q_k(x)|_{x_4=0} = q_k(x)|_{x_4=T} = 0. \qquad (83)$$

Here we give a short comment about the color factor of the gauge action. For any gauge action composed of closed and traced loops, it holds that $A=0$. This means that the color matrix part (which is the color factor without the phase factor that depends on x_4),

$$\text{tr}\left[I^{a_1}\cdots I^{a_r}e^{i\mathcal{E}B}+(-1)^r I^{a_r}\cdots I^{a_1}e^{-i\mathcal{E}B}\right], \qquad (84)$$

is time independent. Therefore one does not have to calculate the color matrix for every x_4 when writing down the vertex in computer code at the Feynman diagram calculation stage. Before the time loop x_4, one can store the color matrix in somewhere, and then in the time loop it can be called. This is realized by virtue of separating the color factor into two parts, the information about lists and the background field.

4.2 Time-momentum space representation

The Fourier transformation for the quantum field is given by

$$q^a_\mu(x) = \frac{1}{L^3}\sum_{\mathbf{p}} e^{i\mathbf{p}\cdot(\mathbf{x}+\frac{\hat{\mu}}{2}\sigma_\mu)}\tilde{q}^a_\mu(\mathbf{p},x_4), \qquad (85)$$

where the spatial momenta are given by $p_k = 2\pi n_k/L$ with $n_k = 0,...,L-1$. An expansion of the action in terms of the coupling g_0 in the time-momentum space is given by

$$\begin{aligned}
S_{\text{G}} &= \sum_{r=2}^{\infty}\frac{(g_0)^{r-2}}{r!}\left(\frac{1}{L^3}\right)^{r-1}\sum_{\mathbf{p}_1,\cdots,\mathbf{p}_r}\delta_{\mathbf{p}_1+\cdots+\mathbf{p}_r,2\pi\mathbf{n}} \\
&\quad \sum_{\mu_1,\cdots,\mu_r}\sum_{t_1,\cdots,t_r}\sum_{a_1,\cdots,a_r} V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(\mathbf{p}_1,\cdots,\mathbf{p}_r;t_1,\cdots,t_r)\prod_{j=1}^{r}\tilde{q}^{a_j}_{\mu_j}(-\mathbf{p}_j,t_j). \qquad (86)
\end{aligned}$$

Similar way to the case in the coordinate space, the symmetrized vertex is given by the partially symmetrized vertex

$$V^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(\mathbf{p}_1,\cdots,\mathbf{p}_r;t_1,\cdots,t_r)=\frac{1}{r!}\sum_{\sigma\in\mathcal{S}_r/\mathcal{Z}_r}\sigma\cdot Y'^{a_1\cdots a_r}_{\mu_1\cdots\mu_r}(\mathbf{p}_1,\cdots,\mathbf{p}_r;t_1,\cdots,t_r). \tag{87}$$

We also have written python code for the time-momentum representation. The list structure for the time-momentum space is a little bit different from the one in the coordinate space. The differences occur in the coordinate of the link variables only, $x\to(t,\mathbf{v})$, where $t=x_4$ and $\mathbf{v}$ is a spatial component of mid-point of the link variables in two lattice units. The list for eq.(66) is given by

$$\begin{aligned}L^{(r)}[i] &= \{(\mu_1[i],\cdots,\mu_r[i]),((t_1[i],\mathbf{v}_1[i]),\cdots,(t_r[i],\mathbf{v}_r[i]),0,0,0,\\ &\quad A[i],B[i],(C_1[i],\cdots,C_r[i]),(D_1[i],\cdots,D_r[i]);f[i]\}.\end{aligned} \tag{88}$$

By making use of the propagator in the time-momentum space

$$\langle\tilde{q}^a_\mu(\mathbf{p},x_0)\tilde{q}^b_\nu(\mathbf{q},y_0)\rangle_0=\delta_{b\bar{a}}L^3\delta_{\mathbf{p}+\mathbf{q},\mathbf{0}}D^a_{\mu\nu}(\mathbf{p},x_0,y_0), \tag{89}$$

the Big Mac diagram in the space is written as

$$\begin{aligned}\text{BigMac} &= \frac{1}{12L^3}\sum_{n(\text{color loop})}^{52}\sum_{i,j(\text{python loop})}^{396}\sum_{t,s}\sum_{\mathbf{p}_1,\mathbf{p}_2}\\ &\quad\frac{1}{3!}f[i]\mathcal{C}_{\rm G}^{a_1[n]a_2[n]a_3[n]}(t,A[i],B[i],C[i],D[i])e^{-\frac{i}{2}(\mathbf{p}_1\cdot\mathbf{v}_1[i]+\mathbf{p}_2\cdot\mathbf{v}_2[i]-\mathbf{q}\cdot\mathbf{v}_3[i])}\\ &\quad\times\frac{1}{3!}f[j]\mathcal{C}_{\rm G}^{\bar{a}_1[n]\bar{a}_2[n]\bar{a}_3[n]}(s,A[j],B[j],C[j],D[j])e^{\frac{i}{2}(\mathbf{p}_1\cdot\mathbf{v}_1[j]+\mathbf{p}_2\cdot\mathbf{v}_2[j]-\mathbf{q}\cdot\mathbf{v}_3[j])}\\ &\quad\times D^{\bar{a}_1[n]}_{\mu_1[j]\mu_1[i]}(\mathbf{p}_1,t+dt_1[j],s+dt_1[i])\\ &\quad\times D^{\bar{a}_2[n]}_{\mu_2[j]\mu_2[i]}(\mathbf{p}_2,t+dt_2[j],s+dt_2[i])\\ &\quad\times D^{a_3[n]}_{\mu_3[i]\mu_3[j]}(\mathbf{q},t+dt_3[i],s+dt_3[j]),\end{aligned} \tag{90}$$

where $q_k=2\pi m_k/L$ with $m_k=n_{1k}+n_{2k}$ mod L for $p_{ik}=2\pi n_{ik}/L$ $(i=1,2)$. We have used the relation,

$$D^a_{\mu\nu}(\mathbf{p},x_0,y_0)=D^{\bar{a}}_{\nu\mu}(-\mathbf{p},y_0,x_0). \tag{91}$$

Note that the computational effort scales with T^2L^6 which is more demanding compared to the coordinate space by L^3. Even though the cost for the Fourier transformation of the propagator in eq.(70) scales proportional to T^2L^6, the prefactor is negligible compared to the sum in the diagram whose factor is $52\times(396)^2$. This shows the benefit of using the coordinate space for the computation of bubble-type diagrams [5].

5 Applications

In this section, we show some applications of the algorithm, especially we consider the one-loop computation of the SF coupling. From this computation we can check that the algorithm works properly and furthermore we can investigate more complicated gauge actions, like improved gauge actions including chair type and parallelogram loops.

5.1 Improved gauge actions

In this subsection, we summarize the definition of the improved gauge actions including six-link loops. More details about the definition can be found in [11]. The improved gauge action for the SF is given by

$$S_{\rm G} = \frac{2}{g_0^2}\sum_{i=0}^{3}\sum_{\mathcal{C}\in\mathcal{S}_i} W_i(\mathcal{C},g_0^2){\rm Retr}\left[1-U(\mathcal{C})\right], \tag{92}$$

where $U(\mathcal{C})$ is a parallel transporter along a loop $\mathcal{C}$. Since we are taking the real part, loops $\mathcal{C}$ which differ by orientation only are considered equal. $\mathcal{S}_i$ denotes sets of loops $\mathcal{C}$ on the lattice as given in Figure 2 together with the corresponding factors c_i, which are normalized by $c_0+8c_1+16c_2+8c_3=1$. We adopt the weight factors for "Choice B" in Ref. [11]. For the plaquette, this means

$$W_0(\mathcal{C},g_0^2) = \begin{cases} c_{\rm s}(g_0^2) & \text{if } \mathcal{C} \text{ lie on one of the boundaries,}\\ c_0 c_{\rm t}^P(g_0^2) & \text{if } \mathcal{C} \text{ just touch one of the boundaries,}\\ c_0 & \text{otherwise,}\end{cases} \tag{93}$$

and for rectangular loops

$$W_1(\mathcal{C},g_0^2) = \begin{cases} 0 & \text{if } \mathcal{C} \text{ lie completely on one of the boundaries,}\\ c_1 c_{\rm t}^R(g_0^2) & \text{if } \mathcal{C} \text{ have exactly 2 links on a boundaries,}\\ c_1 & \text{otherwise,}\end{cases} \tag{94}$$

with the perturbative expansion of the boundary counter term [12]

$$c_0 c_{\rm t}^P(g_0^2) = c_0(1+c_{\rm t}^{P(1)}g_0^2+O(g_0^4)), \tag{95}$$

$$c_1 c_{\rm t}^R(g_0^2) = c_1(3/2+c_{\rm t}^{R(1)}g_0^2+O(g_0^4)). \tag{96}$$

The tree values of the boundary counter terms were given in Ref. [11]. The one-loop coefficients for the improved gauge actions composed from plaquette loops and rectangular loops (Iwasaki, Symanzik and DBW2[6]) were already determined in Ref. [12] with a constraint

$$c_{\rm t}^{R(1)} = 2c_{\rm t}^{P(1)}. \tag{97}$$

As discussed in the reference, this constraint is convenient when we deal with one-loop O(a) improvement, and we assume this in the following. As we shall see later, the remaining one-loop values $c_{\rm t}^{P(1)}$ will be fixed for the other improved gauge actions including the chair and

[6]Note that the value of c_1 for the DBW2 here is a bit different from that in [12].

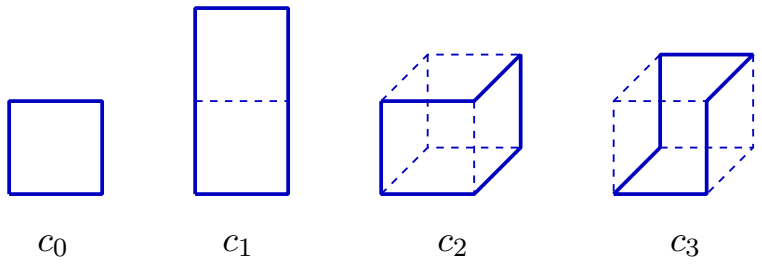

Figure 2: Loops together with corresponding factors.

parallelogram loops. The weight factors for the chair and parallelogram loops ($i = 2, 3$) are given by

$$W_i(\mathcal{C}, g_0^2) = \begin{cases} 0 & \text{if } \mathcal{C} \text{ lie completely on one of the boundaries,} \\ c_i & \text{otherwise.} \end{cases} \tag{98}$$

Gauge actions including the chair and parallelogram loops are hardly used in current simulations, but here we consider such actions for comparison with the old perturbation theory results. We deal with seven gauge actions whose name and values of c_i are given in Table 2.

5.2 One-loop results for the SF coupling

We perform the one-loop computation of the SF coupling in pure SU(3) gauge theory for the improved gauge actions. We set $T = L$ as usual. The SF coupling is defined [4] by

$$\bar{g}^2_{\rm SF}(L) = k \left(\frac{\partial \Gamma}{\partial \eta} \right)^{-1} \Bigg|_{\eta=\nu=0}, \tag{99}$$

with $\Gamma = -\ln \mathcal{Z}[C, C']$ and a normalization

$$k = 12(L/a)^2 \left[c_0 \{ \sin(2\gamma) + \sin(\gamma) \} + 4(c_1 + 2c_2 + c_3) \{ \sin(4\gamma) + \sin(2\gamma) \} \right], \tag{100}$$

where γ is given in eq.(133).

The coupling constant has the perturbative expansion

$$\bar{g}^2_{\rm SF}(L) = g_0^2 + m_1^{(1)}(L/a) g_0^4 + O(g_0^6), \tag{101}$$

and the one-loop coefficient is given by

$$m_1^{(1)}(L/a) = m_1^{(0)}(L/a) - 2c_{\rm t}^{P(1)} R(L/a) a/L, \tag{102}$$

$$R(L/a) = \frac{c_0 \{ \sin(2\gamma) + \sin(\gamma) \} + 4c_1 \{ \sin(4\gamma) + \sin(2\gamma) \}}{c_0 \{ \sin(2\gamma) + \sin(\gamma) \} + 4(c_1 + 2c_2 + c_3) \{ \sin(4\gamma) + \sin(2\gamma) \}}. \tag{103}$$

$m_1^{(0)}(L/a)$ is the one-loop coefficient which is computed with the tree level value of the boundary counter terms, that is, the tree level O(a) improved coefficient. On the other hand, $m_1^{(1)}(L/a)$

Action	c_1	c_2	c_3	A_0	A_1
Plaquette	0	0	0	0.3682819(6)	−0.1779(3)
Wilson RG	−0.252	0	−0.170	−0.2184103(8)	0.2987(3)
Iwasaki	−0.331	0	0	−0.204903(3)	0.304(2)
DBW2	−1.4088	0	0	−0.62816(5)	0.90(2)
Symanzik	−1/12	0	0	0.1361508(3)	−0.0060(1)
Symanzik II	−1/12	1/16	−1/16	0.1403119(5)	0.0216(2)
Symanzik III	−1/12	−0.1	0.1	0.1491183(6)	−0.0745(3)

Table 2: The coefficients A_0 and A_1 of asymptotic expansion of the one-loop coefficient for the various gauge actions.

is a one-loop O(a) improved one which is expected to have no a/L term. What we compute actually is $m_1^{(0)}(L/a)$ in a range $L/a = 6, ..., 64$. The contribution is separated into two parts, the gauge field and the ghost parts,

$$m_1^{(0)} = -\frac{1}{2k}\sum_{a,\mathbf{p}} \mathrm{Tr}\left[(K^a(\mathbf{p}))^{-1}\frac{\partial K^a(\mathbf{p})}{\partial \eta}\right] + (\text{ghost contribution}). \tag{104}$$

The inverse propagator $K^a(\mathbf{p})$ is a sum of the two-point vertex $V_{\mu\nu}^{ab}(\mathbf{p}, -\mathbf{p}; t, s)$ in eq.(86) and the conventional gauge fixing term [4]. We use the ghost action in the previous reference. The η derivative of $K^a(\mathbf{p})$ can be build numerically from the python output as shown in Appendix G. In the one-loop computation we adopt the time-momentum representation which is still advantageous at this order. The trace is taken over the Lorentz and time indices. $c_{\mathrm{P}}^{(1)}$ in eq.(102) can be chosen to cancel the linear term of a/L in $m_1^{(0)}(L/a)$, and then $m_1^{(1)}(L/a)$ turns out to be the O(a) improved quantity.

By analyzing the tree level O(a) improved one-loop results following [13], we obtain the first several coefficients of the Symanzik expansion form,

$$m_1^{(0)}(L/a) = A_0 + B_0 \ln(a/L) + A_1 a/L + B_1 a/L \ln(a/L) + O((a/L)^2). \tag{105}$$

B_0 should be $2b_0$, which is the universal one-loop coefficient of the beta function. We check that $|B_0/(2b_0) - 1| < 10^{-4}$ holds for all improved gauge actions, which we calculate. We also observe that $|B_1| < 10^{-2}$ for all cases, and $B_1 = 0$ is a signal for the achievement of the tree level O(a) improvement. By assuming that $B_0 = 2b_0$ and $B_1 = 0$ we obtain the values of A_0 and A_1, which are given in Table 2. Their values for the case of the plaquette action [4] (the explicit value of A_0 is shown in [13]), and rectangular type actions [12], (Iwasaki and Symanzik) are already known and we confirmed the values to ensure the consistency.

Let us discuss the ratio of the Lambda parameters. Actually there are several old perturbation theory results of the ratio of the lambda parameters between the plaquette gauge action and the improved gauge actions, therefore we can compare with our results. Before taking a

continuum limit, we have to perform the renormalization. We introduce a renormalized coupling $\bar{g}_{\rm Lat}(\mu)$ through

$$\begin{aligned} g_0^2 &= \bar{g}^2_{\rm Lat}(\mu) + z_1(a\mu)\bar{g}^4_{\rm Lat}(\mu) + O(\bar{g}^6_{\rm Lat}(\mu)), & (106)\\ z_1(a\mu) &= 2b_0 \ln(a\mu). & (107) \end{aligned}$$

By substituting eq.(106) into the expression of $\bar{g}^2_{\rm SF}(L)$ of eq.(101) with eq.(105), we obtain

$$\bar{g}^2_{\rm SF}(L) = \bar{g}^2_{\rm Lat}(\mu) + \bar{g}^4_{\rm Lat}(\mu)\left[A_0 + B_0 \ln(a/L) + 2b_0 \ln(a\mu) + O(a/L)\right] + O(\bar{g}^6_{\rm Lat}(\mu)). \quad (108)$$

By noting $B_0 = 2b_0$ and setting $\mu = 1/L$, we can cancel the log divergence. Then we can take the continuum limit

$$\bar{g}^2_{\rm SF}(L) = \bar{g}^2_{\rm Lat}(1/L) + A_0\bar{g}^4_{\rm Lat}(1/L) + O(\bar{g}^6_{\rm Lat}(1/L)). \quad (109)$$

The ratio of the lambda parameters is given in terms of A_0

$$\frac{\Lambda_{\rm Lat}}{\Lambda_{\rm SF}} = \exp[-A_0/(2b_0)]. \quad (110)$$

From this, we can compose a ratio

$$\frac{\Lambda_{\rm impr}}{\Lambda_{\rm Plaq}} = \frac{\Lambda_{\rm impr}/\Lambda_{\rm SF}}{\Lambda_{\rm Plaq}/\Lambda_{\rm SF}} = \exp[(A_0^{\rm Plaq} - A_0^{\rm impr})/(2b_0)]. \quad (111)$$

The resulting values are given in Table 3, we observe rough consistency with the old values. Another ratio of the lambda parameters in the "lattice scheme" and the $\overline{\rm MS}$ scheme is given by

$$\begin{aligned} \frac{\Lambda_{\rm Lat}}{\Lambda_{\overline{\rm MS}}} &= \frac{\Lambda_{\rm Lat}}{\Lambda_{\rm SF}}\frac{\Lambda_{\rm SF}}{\Lambda_{\overline{\rm MS}}} = \exp[-(A_0 + c_1/(4\pi))/(2b_0)], & (112)\\ c_1|_{N_{\rm f}=0} &= 1.255621(2), & (113) \end{aligned}$$

where c_1 is taken from the two-loop SF coupling paper [13], its value for the several gauge actions is given in Table 4. This table may be useful for future references.

From the resulting value of A_1 we can determine the one-loop coefficient of the O(a) boundary counter term $c_{\rm t}^{P(1)}$. The improvement condition yields

$$c_{\rm t}^{P(1)} = \frac{A_1}{2(c_0 + 8c_1)}. \quad (114)$$

Together with eq.(97), one can achieve one-loop O(a) improvement.

5.3 Relative deviation of the step scaling function

Finally, we investigate the relative deviation of the step scaling function (SSF) for the various gauge actions. The SSF for the running coupling [21] is defined by

$$\sigma(u) = \bar{g}^2_{\rm SF}(2L), \quad u = \bar{g}^2_{\rm SF}(L). \quad (115)$$

Action(impr.)	our results	[14]	[15]	[16]	[17]	[18]	[19]
Wilson RG	67.4384(5)	–	–	67.97(9)	–	67.6(3)	67.37(14)
Iwasaki	61.207(2)	61.2064	–	–	59.05(1.00)	–	–
DBW2	1277.1(5)	1276.44	–	–	–	–	–
Symanzik	5.29209(3)	5.29210	5.294(4)	5.29(1)	–	5.29(1)	5.29(1)
Symanzik II	5.13636(3)	–	–	–	–	5.13(1)	–
Symanzik III	4.82173(3)	–	–	4.842(2)	–	–	–

Table 3: Status of $\Lambda_{\rm impr}/\Lambda_{\rm Plaq}$ for pure SU(3) gauge theory. The results from Ref. [16, 17] tend to disagree with ours. Other values from [14, 15, 18, 19] are consistent with our results. Note that the value of c_1 in [14] is a little bit different from ours and actually they use $c_1 = -1.4086$.

Action	$\Lambda_{\rm Lat}/\Lambda_{\overline{\rm MS}}$
Plaquette	0.0347109675049892(2)
Wilson RG	2.34086(2)
Iwasaki	2.12455(5)
DBW2	44.33(2)
Symanzik	0.1836938(4)
Symanzik II	0.1782883(4)
Symanzik III	0.1673674(8)

Table 4: $\Lambda_{\rm Lat}/\Lambda_{\overline{\rm MS}}$ of SU(3) gauge theory. For the plaquette gauge action it is taken from Ref. [20], where a direct calculation of the ratio was done. Our results are those for the other gauge actions.

We denote the SSF measured on the lattice with $\Sigma(u, a/L)$. It is expected to converge to the continuum value,

$$\sigma(u) = \lim_{a\to 0} \Sigma(u, a/L). \tag{116}$$

A scaling behavior to the limit can be described by the relative deviation which is defined from

$$\delta(u, a/L) = \frac{\Sigma(u, a/L) - \sigma(u)}{\sigma(u)}. \tag{117}$$

In perturbation theory, it is expanded as

$$\delta(u, a/L) = \delta_1(a/L)u + O(u^2). \tag{118}$$

The one-loop coefficient is given by

$$\delta_1^{(k)}(a/L) = m_1^{(k)}(2L) - m_1^{(k)}(L/a) - 2b_0 \ln 2, \tag{119}$$

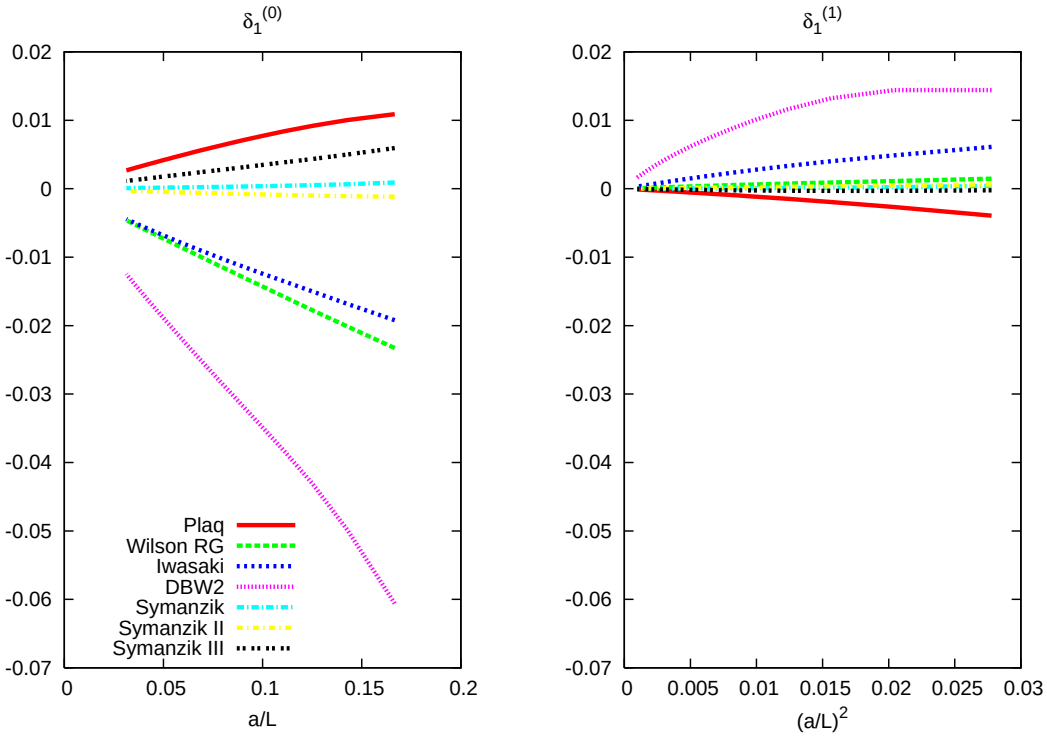

Figure 3: Left panel is for the tree level O(a) improved case, and right one is for the one-loop relative deviation where the O(a) term is removed.

where $k = 0, 1$ denotes the degree of improvement, the results are shown in Figure 3 for the various gauge actions. Symanzik type actions have relatively good scaling behavior, while the renormalization group improved types show rather large cutoff effects at one-loop level.

6 Concluding remarks

In this paper we derive the multiplication algorithm to generate Feynman rules for the SF with an abelian background field. The essential new ingredient for the extension is how to treat the color factor which involves the background field. By making use of the key formulas eq.(30) and eq.(31), which are derived from special properties of the abelian background field (eq.(137)), this difficulty is resolved. We present the multiplication algorithm for the color factors in eq.(46-50). By making use of the algorithm one can obtain vertices to any order for any shape of the parallel transporter in the SF. One has to keep in mind that as shown in Appendix E one needs to perform the remaining symmetrization for the vertices $r \leq 4$ of the gauge action in order to obtain the totally symmetrized vertex at the Feynman diagram calculation stage.

As a first application and to check the correctness of the python code, we calculated the one-loop coefficient of the SF coupling for various gauge actions. We observe consistency by

comparing with previous calculations, and analyze the scaling behavior of the step scaling functions. As a further check we compute the Big Mac diagram for smaller lattices and observe a consistency with the old results. As a further application, we applied the automatic method to a lattice with $L = T \pm a$ which is motivated by considering staggered fermions in the SF [9]. Actually, on that lattice, the time dependence of the background field is not uniform anymore. Therefore we extend our algorithm to this case, the details of the extended algorithm are shown in Appendix D. A preliminary result for the gauge sector by making use of the algorithm was already presented in [9, 22].

It is desirable to go beyond one-loop for actual applications. In that line, the full two-loop computation of the SF coupling is the next target. In this paper, we exclusively consider the gluonic sector as a specific example. Of course, however the multiplication algorithm can be applied for the fermion actions, for example, Wilson fermions and also for the clover term. The algorithm is suited for all link connected objects. Therefore an application for the HQET with highly smearing is also possible.

Some readers, who are interested to use the generalized version of the python script, are invited to ask the author to send it.

Acknowledgments

I would like to thank Alistair Hart to provide me the original python script which was the starting point of my study. I am also grateful to Dirk Hesse, Georg von Hippel, Rainer Sommer and Ulli Wolff for discussions, critical reading and giving comments on the manuscript. I appreciate Peter Weisz sending me the numerical results of the Big Mac diagram. This work is supported in the framework of SFB Transregio 9 of the Deutsche Forschungsgemeinschaft (DFG). I also thank FLAVIAnet for financial support.

A The basis of the Lie algebra $su(3)$

For the Lie algebra $su(3)$ one may choose as a basis

$$T^a = \frac{1}{2i}\lambda^a, \tag{120}$$

with the Gell-Mann matrices λ^a for $a = 1, 2, ..., 8$, such that

$$\mathrm{tr}\left[T^a T^b\right] = -\frac{1}{2}\delta^{ab}, \quad \left[T^a, T^b\right] = f^{abc}T^c. \tag{121}$$

For the SF case, however, it is convenient to choose another basis. New matrices $\tilde{\lambda}^a$ [7, 8] are introduced, which coincide with the Gell-Mann matrices λ^a except for the two diagonal matrices,

$$\tilde{\lambda}^3 = -\frac{1}{2}\lambda^3 + \frac{\sqrt{3}}{2}\lambda^8, \tag{122}$$

$$\tilde{\lambda}^8 = \frac{\sqrt{3}}{2}\lambda^3 + \frac{1}{2}\lambda^8. \tag{123}$$

The matrices $\tilde{\lambda}^a$ are

$$\tilde{\lambda}^1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}^2 = \begin{pmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},$$
$$\tilde{\lambda}^4 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \qquad \tilde{\lambda}^5 = \begin{pmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{pmatrix},$$
$$\tilde{\lambda}^6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \qquad \tilde{\lambda}^7 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{pmatrix},$$
$$\tilde{\lambda}^3 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \qquad \tilde{\lambda}^8 = \frac{1}{\sqrt{3}} \begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}. \tag{124}$$

After a normalization,

$$\tilde{T}^a = \frac{1}{2i}\lambda^a, \tag{125}$$

these matrices may be used to define a new basis I^a, which is given by

$$I^1 = T_+ = \frac{1}{\sqrt{2}}(\tilde{T}^1 + i\tilde{T}^2), \qquad I^2 = T_- = \frac{1}{\sqrt{2}}(\tilde{T}^1 - i\tilde{T}^2),$$
$$I^4 = U_+ = \frac{1}{\sqrt{2}}(\tilde{T}^4 + i\tilde{T}^5), \qquad I^5 = U_- = \frac{1}{\sqrt{2}}(\tilde{T}^4 - i\tilde{T}^5),$$
$$I^6 = V_+ = \frac{1}{\sqrt{2}}(\tilde{T}^6 + i\tilde{T}^7), \qquad I^7 = V_- = \frac{1}{\sqrt{2}}(\tilde{T}^6 - i\tilde{T}^7), \tag{126}$$

for the non–diagonal matrices and

$$I^3 = \tilde{T}^3, \qquad I^8 = \tilde{T}^8, \tag{127}$$

for the diagonal ones. For this basis, one has

$$I^{a\dagger} = -I^{\bar{a}}, \tag{128}$$

where $\bar{1} = 2$, $\bar{4} = 5$, $\bar{6} = 7$, and $\bar{2} = 1$ and so on. For the diagonal matrices, one has $\bar{3} = 3$ and $\bar{8} = 8$. The normalization is chosen such that

$$\mathrm{tr}\left[I^a I^b\right] = -\frac{1}{2}\delta^{b\bar{a}}. \tag{129}$$

B The boundary fields and the constant color electric background field

In the basis chosen in Appendix A, the boundary fields are expressed as

$$\begin{aligned} C &= \frac{i}{L}\begin{pmatrix} \eta-\frac{\pi}{3} & 0 & 0 \\ 0 & \eta(-\frac{1}{2}+\nu) & 0 \\ 0 & 0 & -\eta(\frac{1}{2}+\nu)+\frac{\pi}{3} \end{pmatrix} \\ &= \frac{i}{L}\left[(\eta\nu-\frac{\pi}{6})\tilde{\lambda}^3+\frac{\sqrt{3}}{2}(\eta-\frac{\pi}{3})\tilde{\lambda}^8\right], \end{aligned} \tag{130}$$

$$\begin{aligned} C' &= \frac{i}{L}\begin{pmatrix} -\eta-\pi & 0 & 0 \\ 0 & \eta(\frac{1}{2}+\nu)+\frac{\pi}{3} & 0 \\ 0 & 0 & \eta(\frac{1}{2}-\nu)+\frac{2\pi}{3} \end{pmatrix} \\ &= \frac{i}{L}\left[(\eta\nu-\frac{\pi}{6})\tilde{\lambda}^3-\frac{\sqrt{3}}{2}(\eta+\pi)\tilde{\lambda}^8\right]. \end{aligned} \tag{131}$$

The constant color electric background field $\mathcal{E}$ is proportional to I^8

$$\mathcal{E} = -\gamma\begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix} = -\gamma\sqrt{3}\tilde{\lambda}^8 = -2\sqrt{3}i\gamma I^8, \tag{132}$$

with

$$\gamma = \frac{1}{LT}\left(\eta+\frac{\pi}{3}\right). \tag{133}$$

The η derivative of $\mathcal{E}$ is given by

$$\frac{\partial\mathcal{E}}{\partial\eta} = -2\sqrt{3}i\frac{\partial\gamma}{\partial\eta}I^8 = \frac{-2\sqrt{3}i}{LT}I^8. \tag{134}$$

C The Cartan sub-algebra

Let us assume that H_i for $i = 1, \cdots, m$ (m is the rank of the algebra) are hermitian and elements of the Cartan sub-algebra. They commute,

$$[H_i, H_j] = 0. \tag{135}$$

The generators E_a of the original algebra satisfy

$$[H_i, E_a] = \mu_{ai}E_a, \tag{136}$$

where μ_{ai} are the roots, which are the weights of the adjoint representation. By making use of eq.(136), one can see that

$$e^{i\sum_j h_j H_j}E_a e^{-i\sum_j h_j H_j} = E_a e^{i\sum_j h_j\mu_{aj}}, \tag{137}$$

a	1	2	3	4	5	6	7	8
μ_{a1}	$-\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$-\frac{1}{2}$	1	-1	0
μ_{a2}	$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{2}$	0	$\frac{\sqrt{3}}{2}$	$-\frac{\sqrt{3}}{2}$	0	0	0

Table 5: The roots μ_{ai} for SU(3) case.

with real coefficients h_i. An E_a sandwiched between group elements generated by the Cartan algebra does not mix with other E_b ($b \neq a$), but turns out to be the E_a itself multiplied a phase factor.

In our case of $su(3)$, the Cartan generators are identified as $H_1 = iI^3$, $H_2 = iI^8$, and the other generators as $E_a = iI^a$. The values of the roots for the $su(3)$ case are shown in Table 5. In terms of the Cartan generator, the background field in eq.(20) can be written as

$$V(x_4) = e^{i\sum_j h_j(x_4)H_j}, \tag{138}$$

where the coefficients $h_j(x_4)$ can be read off from the definition of $V(x_4)$. By making use of the above formula, one can see that the phase $\phi_a(x_4)$ in eq.(30) is given in terms of the coefficients h_j and the roots μ_{aj}

$$\phi_a(x_4) = \sum_j h_j(x_4)\mu_{aj}. \tag{139}$$

D Multiplication for a background field with an arbitrary time dependence

In this section, we provide a multiplication algorithm for the color factor for an abelian background field which has arbitrary time dependence. We start from a general form of the color factor of order r for this case,

$$\begin{aligned}
&\mathcal{C}^{a_1\cdots a_r}(t_{\min}, t_{\max}, A, D) \\
&= \left[I_{a_1}\cdots I_{a_r} V(t_{\min})^{A_{t_{\min}}} V(t_{\min}+1)^{A_{t_{\min}+1}} \cdots V(t_{\max})^{A_{t_{\max}}}\right] \times \\
&\quad \times e^{\frac{i}{2}\sum_{u=1}^{r}(D_{u,t_{\min}}\phi_{a_u}(t_{\min}) + D_{u,t_{\min}+1}\phi_{a_u}(t_{\min}+1) + \cdots + D_{u,t_{\max}}\phi_{a_u}(t_{\max}))}.
\end{aligned} \tag{140}$$

For a certain configuration $\{a_1, \cdots, a_r\}$, the color factor is specified by the set $\{t_{\min}, t_{\max}, A, D\}$. $t_{\min}$ and $t_{\max}$ are defined in Figure 4. The explicit forms of A and D are given by

$$A = (A_{t_{\min}}, A_{t_{\min}+1}, \cdots, A_{t_{\max}}), \tag{141}$$

$$D = (D_{t_{\min}}, D_{t_{\min}+1}, \cdots, D_{t_{\max}}), \tag{142}$$

$$D_t = (D_{1t}, D_{2t}, \cdots, D_{rt}), \text{ for } t_{\min} \leq t \leq t_{\max}. \tag{143}$$

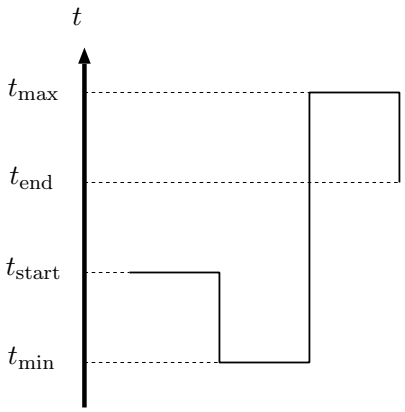

Figure 4: A line which show $t_{\max}$, $t_{\min}$, $t_{\rm start}$ and $t_{\rm end}$.

Note that all elements of A, D are integer valued. Let us see an example of (A, D) for the color factor for the single link variable $U(x,\mu)^m$. For $x = 0$ and $\mu = 4$ ($t_{\min} = 0$, $t_{\max} = 1$),

$$\begin{aligned}(A,D) &= ((A_0,A_1),(D_0,D_1)) \\ &= ((0,0),(\underbrace{(0,0,...,0)}_{r \text{ terms}},\underbrace{(0,0,...,0)}_{r \text{ terms}})).\end{aligned} \tag{144}$$

For $x = 0$ and $\mu = 1, 2, 3$ ($t_{\min} = 0$, $t_{\max} = 0$),

$$\begin{aligned}(A,D) &= (A_0,D_0) \\ &= (0,\underbrace{(0,0,...,0)}_{r \text{ terms}}).\end{aligned} \tag{145}$$

Even for this case, the color factors are closed under the multiplication ($r = r_1 + r_2$),

$$\begin{aligned}&\mathcal{C}^{a_1\cdots a_{r_1}}(t^{(1)}_{\min},t^{(1)}_{\max},A^{(1)},D^{(1)})\mathcal{C}^{a_{r_1+1}\cdots a_{r_1+r_2}}(t^{(2)}_{\min},t^{(2)}_{\max},A^{(2)},D^{(2)}) \\ =\ &\mathcal{C}^{a_1\cdots a_r}(t_{\min},t_{\max},A,D).\end{aligned} \tag{146}$$

So we found the following multiplication algorithm. First, the resulting $t_{\min}$ and $t_{\max}$ are given by,

$$t_{\min} = \min(t^{(1)}_{\min}, t^{(2)}_{\min}), \tag{147}$$

$$t_{\max} = \max(t^{(1)}_{\max}, t^{(2)}_{\max}). \tag{148}$$

Before operating A and D, we have to pad zeros, from the left

$$\begin{aligned}A^{(1)} \longrightarrow \tilde{A}^{(1)} &= (\tilde{A}^{(1)}_{t_{\min}},\cdots,\tilde{A}^{(1)}_{t_{\max}}) \\ &= \begin{cases} (\underbrace{0,0,...,0}_{(t^{(1)}_{\min}-t_{\min}) \text{ terms}},A^{(1)}_{t^{(1)}_{\min}},...,A^{(1)}_{t^{(1)}_{\max}}), & \text{for } t^{(1)}_{\min} > t_{\min}, \\ (A^{(1)}_{t^{(1)}_{\min}},...,A^{(1)}_{t^{(1)}_{\max}}), & \text{for } t^{(1)}_{\min} = t_{\min}, \end{cases}\end{aligned} \tag{149}$$

where we note that the size of $\tilde{A}^{(1)}$ is $t_{\max} - t_{\min} + 1$, while that of $A^{(1)}$ is $t^{(1)}_{\max} - t^{(1)}_{\min} + 1$. Also from the right,

$$\begin{aligned} A^{(1)} \longrightarrow \tilde{A}^{(1)} &= (\tilde{A}^{(1)}_{t_{\min}}, \cdots, \tilde{A}^{(1)}_{t_{\max}}) \\ &= \begin{cases} (A^{(1)}_{t^{(1)}_{\min}}, ..., A^{(1)}_{t^{(1)}_{\max}}, \underbrace{0, 0, ..., 0}_{(t_{\max} - t^{(1)}_{\max}) \text{ terms}}), & \text{for } t^{(1)}_{\max} < t_{\max}, \\ (A^{(1)}_{t^{(1)}_{\min}}, ..., A^{(1)}_{t^{(1)}_{\max}}), & \text{for } t^{(1)}_{\max} = t_{\max}. \end{cases} \end{aligned} \tag{150}$$

In a similar way, we can pad the others with zeros and obtain $\tilde{A}^{(2)}$ and $\tilde{D}^{(1,2)}$. Then the operation is done for the tilde objects,

$$A_t = \tilde{A}^{(1)}_t + \tilde{A}^{(2)}_t, \tag{151}$$

$$D_t = \underbrace{(\underbrace{\tilde{D}^{(1)}_{1t}, \cdots, \tilde{D}^{(1)}_{r_1 t}}_{r_1 \text{ terms}}, \underbrace{\tilde{D}^{(2)}_{1t} + 2\tilde{A}^{(1)}_t, \cdots, \tilde{D}^{(2)}_{r_2 t} + 2\tilde{A}^{(1)}_t}_{r_2 \text{ terms}})}_{r \text{ terms}}, \tag{152}$$

for $t_{\min} \leq t \leq t_{\max}$. For the gauge action, the symmetrization and the reduction of the number of lists can also be applied in this case in a similar way to Appendix E and F.

E Partial symmetrization of the vertex at the python level

For the gauge action (traced and real), the summation over all possible permutation in eq.(87) can be done partially at the python level, that is, by creating new lists corresponding to a symmetrized vertex. It is maybe worth starting from the vanishing background field [1, 2]. The color factor in this case ($\mathcal{E} = 0$, $\phi_a = \phi'_a = 0$) is given by

$$\mathcal{C}^{a_1 \cdots a_r}_{\mathrm{G}} = \mathrm{tr}[I^{a_1} \cdots I^{a_r} + (-1)^r I^{a_r} \cdots I^{a_1}]. \tag{153}$$

This is an eigenstate of the inversion and cyclic permutations.

$$\sigma \cdot \mathcal{C}_{\mathrm{G}} = \chi_r(\sigma) \mathcal{C}_{\mathrm{G}}, \tag{154}$$

where the eigenvalues are given by

$$\chi_r(\sigma) = \begin{cases} 1, & \text{for cyclic permutations,} \\ (-1)^r, & \text{for inversion permutation.} \end{cases} \tag{155}$$

By using these properties, the symmetrized vertex can be written as[7]

$$V^{a_1 \cdots a_r}_{\mu_1 \cdots \mu_r}(\mathbf{p}_1, \cdots, \mathbf{p}_r; t_1, \cdots, t_r) = \frac{1}{r!} \sum_{\sigma \in \mathcal{S}_r / \mathcal{Z}_r} \sigma \cdot \mathcal{C}^{a_1 \cdots a_r}_{\mathrm{G}} \sigma \cdot Y'_{\mu_1 \cdots \mu_r}(\mathbf{p}_1, \cdots, \mathbf{p}_r; t_1, \cdots, t_r), \tag{156}$$

[7] Here we show the time-momentum space representation, but the following discussion is also valid for the coordinate space case.

with

$$Y'_{\mu_1\cdots\mu_r}(\mathbf{p}_1,\cdots,\mathbf{p}_r;t_1,\cdots,t_r)=\sum_{\sigma\in\mathcal{Z}_r}\chi_r(\sigma)\sigma\cdot Y_{\mu_1\cdots\mu_r}(\mathbf{p}_1,\cdots,\mathbf{p}_r;t_1,\cdots,t_r). \tag{157}$$

In Ref. [2], it is described how to perform the summation in eq.(157) at the python level. We have seen the nice properties of the color factor of the gauge action in the case of a vanishing background field. At first sight, however, it is nontrivial that the color factor for SF has such properties. We will discuss this issue in the following subsections, and explain how to obtain a set for the partially symmetrized vertex.

E.1 The inversion permutation for the color factor

First, let us see the inversion permutation for the color factor of the gauge action in the SF. An inverted color factor is written by (remember that $A=0$ for the gauge action)

$$\begin{aligned}\sigma_{\rm inv}\cdot\mathcal{C}_{\rm G}^{a_1a_2\cdots a_{r-1}a_r}(x_4,0,B,C,D) &= \mathcal{C}_{\rm G}^{a_ra_{r-1}\cdots a_2a_1}(x_4,0,B,C,D)\\ &= (-1)^r\mathcal{C}_{\rm G}^{a_1a_2\cdots a_{r-1}a_r}(x_4,0,\tilde{B},\tilde{C},\tilde{D}),\end{aligned} \tag{158}$$

with

$$\tilde{B} = -B, \tag{159}$$

$$(\tilde{C}_1,\tilde{C}_2,\cdots,\tilde{C}_{r-1},\tilde{C}_r) = (C_r,C_{r-1},\cdots,C_2,C_1), \tag{160}$$

$$(\tilde{D}_1,\tilde{D}_2,\cdots,\tilde{D}_{r-1},\tilde{D}_r) = (D_r,D_{r-1},\cdots,D_2,D_1). \tag{161}$$

Under the inversion permutation, the color factor is not an eigenstate anymore. However the structure is still stable but with different components $\tilde{B}$, $\tilde{C}$ and $\tilde{D}$.

E.2 Cyclic permutations for the color factor

Secondly, let us consider the cyclic permutation. A cyclically permuted color factor is written as

$$\begin{aligned}\sigma_{\rm cyc}\cdot\mathcal{C}_{\rm G}^{a_1a_2\cdots a_{r-1}a_r}(x_4,0,B,C,D) &= \mathcal{C}_{\rm G}^{a_2a_3\cdots a_ra_1}(x_4,0,B,C,D)\\ &= \mathcal{C}_{\rm G}^{a_1a_2\cdots a_{r-1}a_r}(x_4,0,\hat{B},\hat{C},\hat{D}),\end{aligned} \tag{162}$$

where we have used eq.(31). The explicit forms of the "hat" objects are as follows,

$$\hat{B} = B, \tag{163}$$

$$(\hat{C}_1,\hat{C}_2,\cdots,\hat{C}_{r-1},\hat{C}_r) = (C_r-2B,C_1,C_2,\cdots,C_{r-1}), \tag{164}$$

$$(\hat{D}_1,\hat{D}_2,\cdots,\hat{D}_{r-1},\hat{D}_r) = (D_r,D_1,D_2,\cdots,D_{r-1}). \tag{165}$$

Composite cyclic permutations can be done in a similar way. For $1\le j\le r-1$,

$$\begin{aligned}(\sigma_{\rm cyc})^j\cdot\mathcal{C}_{\rm G}^{a_1\cdots a_r}(x_4,0,B,C,D) &= \mathcal{C}_{\rm G}^{a_{j+1}\cdots a_{r-1}a_ra_1a_2\cdots a_j}(x_4,0,B,C,D)\\ &= \mathcal{C}_{\rm G}^{a_1a_2\cdots a_{r-1}a_r}(x_4,0,\hat{B},\hat{C},\hat{D}),\end{aligned} \tag{166}$$

with

$$\begin{aligned}
\hat{B} &= B, && (167)\\
(\hat{C}_1, \hat{C}_2, \cdots, \hat{C}_r) &= (C_{r-j+1} - 2B, C_{r-j+2} - 2B, \cdots, C_r - 2B, C_1, C_2, \cdots, C_{r-j}), && (168)\\
(\hat{D}_1, \hat{D}_2, \cdots, \hat{D}_r) &= (D_{r-j+1}, D_{r-j+2}, \cdots, D_r, D_1, D_2, \cdots, D_{r-j}). && (169)
\end{aligned}$$

As a further example, we show an opposite cyclic permutation.

$$\begin{aligned}
(\sigma_{\rm cyc})^{-1} \cdot \mathcal{C}_{\rm G}^{a_1 a_2 \cdots a_{r-1} a_r}(x_4, 0, B, C, D) &= \mathcal{C}_{\rm G}^{a_r a_1 \cdots a_{r-2} a_{r-1}}(x_4, 0, B, C, D)\\
&= \mathcal{C}_{\rm G}^{a_1 a_2 \cdots a_{r-1} a_r}(x_4, 0, \hat{B}, \hat{C}, \hat{D}), && (170)
\end{aligned}$$

with

$$\begin{aligned}
\hat{B} &= B, && (171)\\
(\hat{C}_1, \hat{C}_2, \cdots, \hat{C}_{r-1}, \hat{C}_r) &= (C_2, C_3, \cdots, C_r, C_1 + 2B), && (172)\\
(\hat{D}_1, \hat{D}_2, \cdots, \hat{D}_{r-1}, \hat{D}_r) &= (D_2, D_3, \cdots, D_r, D_1). && (173)
\end{aligned}$$

If one uses eq.(194), it is not so difficult to see that

$$(\sigma_{\rm cyc})^{-1} \cdot \mathcal{C}_{\rm G} = (\sigma_{\rm cyc})^{r-1} \cdot \mathcal{C}_{\rm G}. \tag{174}$$

E.3 The partially symmetrized vertex for SF

Let us see how to construct the set for the partially symmetrized vertex for the SF,

$$Y'^{a_1 \cdots a_r}_{\mu_1 \cdots \mu_r}(\mathbf{p}_1, \cdots, \mathbf{p}_r; t_1, \cdots, t_r) = \sum_{\sigma \in \mathcal{Z}_r} \sigma \cdot Y^{a_1 \cdots a_r}_{\mu_1 \cdots \mu_r}(\mathbf{p}_1, \cdots, \mathbf{p}_r; t_1, \cdots, t_r). \tag{175}$$

Starting from lists for the unsymmetrized vertex $L^{(r)}[i]$, one can create corresponding lists in the following way,

$$\begin{array}{ccc}
L^{(r)}[i] & \xrightarrow{cyclic} & (\sigma_{\rm cyc})^j \cdot L^{(r)}[i] \qquad (1 \le j \le r-1)\\
\Big\downarrow inverse & & \\
\sigma_{\rm inv} \cdot L^{(r)}[i] & \xrightarrow{cyclic} & (\sigma_{\rm cyc})^j \cdot \sigma_{\rm inv} \cdot L^{(r)}[i] \qquad (1 \le j \le r-1).
\end{array} \tag{176}$$

Note that in the SF case the cyclic and inversion permutations are not commutative. An algorithm to obtain an inverted list, $\sigma_{\rm inv} \cdot L^{(r)}[i]$, from the original list $L^{(r)}[i]$

$$L^{(r)}[i] \longrightarrow \sigma_{\rm inv} \cdot L^{(r)}[i], \tag{177}$$

is as follows,

$$
\begin{aligned}
(\mu_1, \mu_2, \cdots, \mu_{r-1}, \mu_r) &\longrightarrow (\mu_r, \mu_{r-1}, \cdots, \mu_2, \mu_1), && (178)\\
(t_1, t_2, \cdots, t_{r-1}, t_r) &\longrightarrow (t_r, t_{r-1}, \cdots, t_2, t_1), && (179)\\
(\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{r-1}, \mathbf{v}_r) &\longrightarrow (\mathbf{v}_r, \mathbf{v}_{r-1}, \cdots, \mathbf{v}_2, \mathbf{v}_1), && (180)\\
B &\longrightarrow -B, && (181)\\
(C_1, C_2, \cdots, C_{r-1}, C_r) &\longrightarrow (C_r, C_{r-1}, \cdots, C_2, C_1), && (182)\\
(D_1, D_2, \cdots, D_{r-1}, D_r) &\longrightarrow (D_r, D_{r-1}, \cdots, D_2, D_1), && (183)\\
f &\longrightarrow (-1)^r f. && (184)
\end{aligned}
$$

For a cyclically permuted list $\sigma_{\rm cyc} \cdot L^{(r)}[i]$,

$$
L^{(r)}[i] \longrightarrow \sigma_{\rm cyc} \cdot L^{(r)}[i], \tag{185}
$$

it is given by,

$$
\begin{aligned}
(\mu_1, \mu_2, \cdots, \mu_{r-1}, \mu_r) &\longrightarrow (\mu_2, \mu_3, \cdots, \mu_r, \mu_1), && (186)\\
(t_1, t_2, \cdots, t_{r-1}, t_r) &\longrightarrow (t_2, t_3, \cdots, t_r, t_1), && (187)\\
(\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_{r-1}, \mathbf{v}_r) &\longrightarrow (\mathbf{v}_2, \mathbf{v}_3, \cdots, \mathbf{v}_r, \mathbf{v}_1), && (188)\\
B &\longrightarrow B, && (189)\\
(C_1, C_2, \cdots, C_{r-1}, C_r) &\longrightarrow (C_2, C_3, \cdots, C_r, C_1 + 2B), && (190)\\
(D_1, D_2, \cdots, D_{r-1}, D_r) &\longrightarrow (D_2, D_3, \cdots, D_r, D_1), && (191)\\
f &\longrightarrow f. && (192)
\end{aligned}
$$

By gathering all lists, $\{L^{(r)}[i]\}$, $\{\sigma_{\rm inv} \cdot L^{(r)}[i]\}$, $\{(\sigma_{\rm cyc})^j \cdot L^{(r)}[i] | 1 \leq j \leq r-1\}$ and $\{(\sigma_{\rm cyc})^j \cdot \sigma_{\rm inv} \cdot L^{(r)}[i] | 1 \leq j \leq r-1\}$, the set S for the partially symmetrized vertex can be composed,

$$
\begin{aligned}
S &= \{L^{(r)}[i], \sigma_{\rm cyc} \cdot L^{(r)}[i], \cdots, (\sigma_{\rm cyc})^{r-1} \cdot L^{(r)}[i],\\
&\quad \sigma_{\rm inv} \cdot L^{(r)}[i], \sigma_{\rm cyc} \cdot \sigma_{\rm inv} \cdot L^{(r)}[i], \cdots, (\sigma_{\rm cyc})^{r-1} \cdot \sigma_{\rm inv} \cdot L^{(r)}[i]\}. && (193)
\end{aligned}
$$

F Reduction of the number of lists

One can reduce the number of lists by making use of the properties of the background field. From eq.(30) and (31), we found that the color factor obeys the relation

$$
\begin{aligned}
&\mathcal{C}_{\rm G}^{a_1 \cdots a_r}(x_4, A, B, (C_1, \cdots, C_r), (D_1, \cdots, D_r))\\
&\quad = \mathcal{C}_{\rm G}^{a_1 \cdots a_r}(x_4, A, B, (C_1 + \alpha, \cdots, C_r + \alpha), (D_1 + \beta, \cdots, D_r + \beta)), && (194)
\end{aligned}
$$

for arbitrary real numbers α, β. This equation tells us that one can shift all elements of the $C = (C_1, \cdots, C_r)$ and $D = (D_1, \cdots, D_r)$ to a standard form such that $C_1 \to 0$ and $D_1 \to 0$, that is, choosing $\alpha = -C_1$ and $\beta = -D_1$. After performing the shift for C and D of all lists

in a set, there might be some lists (e.g. two lists $L^{(r)}[i]$ and $L^{(r)}[j]$ with $i \neq j$) which have an identical configuration $C[i] = C[j]$ and $D[i] = D[j]$ (of course the Lorentz index configuration, the coordinate configuration[8], A and B in the two lists should be identical beforehand, but their amplitudes do not have to be so). In this case, we can merge the two lists into a list by summing the amplitude factor $f[i] + f[j]$. In this way one can reduce the number of lists. Actually this achieves a few percents reduction of the size of a set for typical gauge actions.

G η derivative of the vertex

When one calculates the SF coupling, the η derivatives of the vertices are required. We can easily obtain them from the same python output for the usual (non η-derivative) vertices. Since the information of the background field ($V(x_4)$, $\mathcal{E}$, ϕ'_a and $\phi_a(x_4)$ which have η dependence) is encoded in the color factor, we need to consider the color factor only for our purpose. For example, the explicit form of the η derivative of the color factor for the gauge action ($A = 0$) is given by

$$\begin{aligned}
&\frac{\partial \mathcal{C}_{\rm G}^{a_1\cdots a_r}(x_4,0,B,C,D)}{\partial\eta} \\
&= \left(\frac{2\sqrt{3}a^2B}{LT}\right) \mathrm{tr}\left[I^{a_1}\cdots I^{a_r} I^8 e^{ia^2\mathcal{E}B} - (-1)^r I^8 I^{a_r}\cdots I^{a_1} e^{-ia^2\mathcal{E}B}\right] e^{\frac{i}{2}\sum_{u=1}^r (C_u\phi'_{a_u} + D_u\phi_{a_u}(x_4))} \\
&+ \frac{i}{2}\sum_{u=1}^r \left(C_u\partial_\eta\phi'_{a_u} + D_u\partial_\eta\phi_{a_u}(x_4)\right)\mathcal{C}_{\rm G}^{a_1\cdots a_r}(x_4,0,B,C,D), \qquad (195)
\end{aligned}$$

where we have used eq.(134). The information of B, C and D is enough to identify the above term. When writing down the η derivative vertex on a diagram calculation code, one needs additional phases $\partial_\eta\phi'_a$ and $\partial_\eta\phi_a(x_4)$ which are given in Table 1 in analytic forms, and an additional color matrix

$$\mathrm{tr}\left[I^{a_1}\cdots I^{a_r} I^8 e^{ia^2\mathcal{E}B} - (-1)^r I^8 I^{a_r}\cdots I^{a_1} e^{-ia^2\mathcal{E}B}\right]. \qquad (196)$$

This has a form similar to that in eq.(84) of the usual vertex apart from the presence of I^8 and the relative sign between the two terms.

References

[1] M. Lüscher and P. Weisz, *Efficient numerical techniques for perturbative lattice gauge theory computations*, *Nucl. Phys.* **B266** (1986) 309.

[2] A. Hart, G. M. von Hippel, R. R. Horgan and L. C. Storoni, *Automatically generating Feynman rules for improved lattice field theories*, *J. Comput. Phys.* **209** (2005) 340–353 [hep-lat/0411026].

[8] In the momentum space representation, the replacement $x[i] \to (t[i], \mathbf{v}[i])$ is understood.

[3] A. Hart, G. M. von Hippel and R. R. Horgan, *Leptonic widths of heavy quarkonia: S-wave QCD / NRQCD matching coefficients for the electromagnetic vector annihilation current at O(alpha(s) v**2)*, hep-lat/0605007.

[4] M. Lüscher, R. Sommer, P. Weisz and U. Wolff, *A precise determination of the running coupling in the SU(3) Yang-Mills theory*, *Nucl. Phys.* **B413** (1994) 481–502 [hep-lat/9309005].

[5] R. Narayanan and U. Wolff, *Two loop computation of a running coupling lattice Yang-Mills theory*, *Nucl. Phys.* **B444** (1995) 425–446 [hep-lat/9502021].

[6] M. Lüscher, R. Narayanan, P. Weisz and W. Ulli, *The Schrodinger functional: A renormalizable probe for nonabelian gauge theories*, *Nucl. Phys.* **B384** (1992) 168–228 [hep-lat/9207009].

[7] P. Weisz, "csw note (private note)." 1996.

[8] S. Kurth, "The renormalized quark mass in the Schrodinger functional of lattice QCD: A one loop calculation with a nonvanishing background field." 2002.

[9] P. Perez-Rubio and S. Sint, *The SF running coupling with four flavours of staggered quarks*, *PoS* **LATTICE2007** (2006) 249 [0710.0583].

[10] U. Wolff, "Symbolic tools for perturbation theory for the Schrödinger functional." 2007.

[11] S. Aoki, R. Frezzotti and P. Weisz, *Computation of the improvement coefficient csw to 1-loop with improved gluon actions*, *Nucl. Phys.* **B540** (1999) 501–519 [hep-lat/9808007].

[12] S. Takeda, S. Aoki and K. Ide, *A perturbative determination of O(a) boundary improvement coefficients for the Schroedinger functional coupling at 1-loop with improved gauge actions*, *Phys. Rev.* **D68** (2003) 014505 [hep-lat/0304013].

[13] **ALPHA** Collaboration, A. Bode, P. Weisz and U. Wolff, *Two loop computation of the Schroedinger functional in lattice QCD*, *Nucl. Phys.* **B576** (2000) 517–539 [hep-lat/9911018].

[14] A. Skouroupathis and H. Panagopoulos, *Lambda-parameter of lattice QCD with Symanzik improved gluon actions*, *Phys. Rev.* **D76** (2007) 114514 [0709.3239].

[15] P. Weisz and R. Wohlert, *Continuum limit improved lattice action for pure Yang-Mills theory. 2*, *Nucl. Phys.* **B236** (1984) 397.

[16] Y. Iwasaki and S. Sakai, *The lambda parameter for improved lattice gauge actions*, *Nucl. Phys.* **B248** (1984) 441.

[17] Y. Iwasaki and T. Yoshie, *Renormalization group improved action for SU(3) lattice gauge theory and the string tension*, *Phys. Lett.* **B143** (1984) 449.

[18] A. Ukawa and S.-K. Yang, *Asymptotic freedom scales for SU(N) lattice gauge theory with improved action*, *Phys. Lett.* **B137** (1984) 201.

[19] W. Bernreuther, W. Wetzel and R. Wohlert, *Lambda parameters for lattice Yang-Mills actions containing plaquettes with six links*, *Phys. Lett.* **B142** (1984) 407–409.

[20] M. Lüscher and P. Weisz, *Two loop relation between the bare lattice coupling and the ms coupling in pure SU(N) gauge theories*, *Phys. Lett.* **B349** (1995) 165–169 [hep-lat/9502001].

[21] M. Lüscher, P. Weisz and U. Wolff, *A Numerical method to compute the running coupling in asymptotically free theories*, *Nucl. Phys.* **B359** (1991) 221–243.

[22] S. Takeda and U. Wolff, *Automatic generation of vertices for the Schroedinger functional*, *PoS* **LATTICE2007** (2006) 257 [0709.4167].